RL-TR-91-200
Final Technical Report
September 1991

AUTOMATED TESTABILITY DECISION TOOL

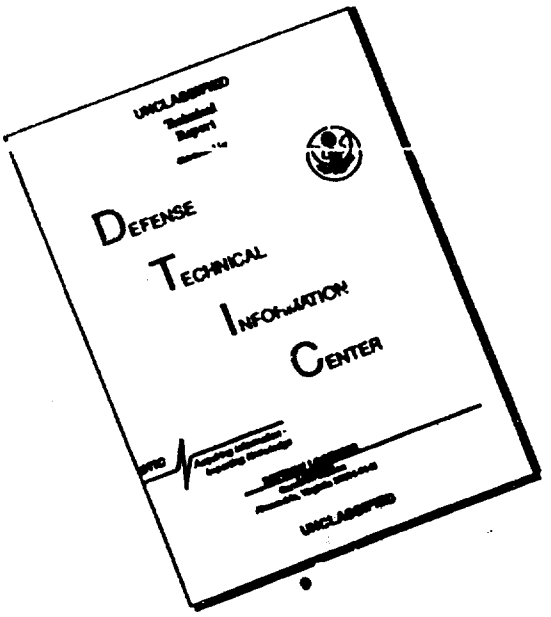
Harris Corporation

Dr. David M. Bellehsen, Brian A. Kelly, Alony Hanania,
et al.

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-91-200 has been reviewed and is approved for publication.

APPROVED: Frank H. Born

FRANK H. BORN
Project Engineer

FOR THE COMMANDER: Raymond C. White

RAYMOND C. WHITE, Colonel, USAF
Director of Reliability & Compatibility

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL(ERSR) Griffiss AFB, NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1991	Final Jul 86 - Sep 89

4. TITLE AND SUBTITLE
AUTOMATED TESTABILITY DECISION TOOL

5. FUNDING NUMBERS
C - F30602-87-D-0185,
Task 0004
PE - 62702F
PR - 2338
TA - J2
WU - C1

6. AUTHOR(S)
Dr. David L. Bellehsen, Brian A. Kelly, Alony Hanania, et al.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Harris Corporation
Government Support Systems Division
6801 Jericho Turnpike
Syosset NY 11791

8. PERFORMING ORGANIZATION REPORT NUMBER
6323-D004

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory (ERSR)
Griffiss AFB NY 13441-5700

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
RL-TR-91-200

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Frank H. Born/ERSR/(315) 330-4726

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This report provides guidelines, mathematical tools and procedures for computing, assessing and allocating testability to a new system design while staying within the confines of the System Engineering Process. These procedures, in turn, will be used by equipment designers to optimize the cost-effectiveness of incorporating testability, via BIT/BITE, ETE or any combination thereof into a weapon system. Specifically, this report provides the following:

- Testability Figures of Merit (TFOMs) used to describe and quantify testability as applied to a weapon system in precise and measurable engineering terms.
- Testability Allocation Methods (TAM) to apportion cost effectively system testability requirements through lower levels of indenture to the Line Replaceable Units (LRUs), and generate subsystem level requirements.

NOTE: Rome Laboratory/RI (formerly Rome Air Development Center/RADC)

14. SUBJECT TERMS
Testability, Requirements Allocation, Test Methods, Computer-Aided Design (CAD)

15. NUMBER OF PAGES
272

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

EXECUTIVE SUMMARY

One of the great needs in the testability discipline is for tools and techniques to enable cost-effective allocation of testability. System level testability requirements must be allocated down to lower levels of assembly where they are meaningful to a designer. The process involved in allocating testability is a complex one for a number of reasons:

1) Testability defies simple definition. Several TFOMs are required to adequately cover all of its aspects for system level specification. As such, Testability Figures Of Merit (TFOMs) are often conflicting or ambiguously defined. Besides this, the criticality of system faults must be an overriding factor, necessitating separate requirements for detection of mission critical or safety critical faults.

2) Interactions between TFOMs are generally not well understood. It is often the case that increases in one category of TFOMs results in decreases in another category of TFOMs. (Eg. Increases in fault detection capability often results in higher false alarm rates).

3) Increases in testability usually have some 'costs' (negative impacts on system weight, power requirements, and speed of operation). It can also have effects on the reliability of the system. Although these impacts are often minor, they must be considered when optimizing the testability on a system.

Contained in this report are details of a study conducted by Harris Corporation for Rome Laboratory (formerly Rome Air Development Center, RADC) to begin development of a computer module for performing the allocation of testability requirements considering the factors discussed above. The specific contributions toward this end are:

Section 2: TFOM DEVELOPMENT
Over 100 Testability Figures of Merit were considered in the effort to determine an optimum set of TFOMs for system level specification. The resulting six TFOMs were chosen after applying a series of screens to the larger group. The screens were used to ensure maximum coverage of testability characteristics, orthogonality of the resulting set, minimum ambiguity of selected TFOMs, etc. Also included in this section are analytical techniques for translating the selected TFOMs down to lower levels of assembly.

Section 3: TAM Development
The Testability Allocation Module (TAM) developed in this effort utilizes an Augmented Lagrangian approach to solving the many equations involved in optimizing the allocation. Also included in this section is a formulation of the problem and an algorithm for its solution. The problem can be generalized as "Maximize testability given system 'cost functions'", or equivalently, "minimize total 'cost' given system testability/design requirements".

Section 4: TFOM/TAM Integration
Models detailing the complex interactions between diagnostic system performance parameters, as measured by the TFOMs and system reliability availability maintainability and life cycle cost, are presented. From these relationships procedures are developed to integrate the cost functions (objectives and constraints) and the allocation methodology. Section 4 also presents application examples showing optimization of testability subject to 'cost' constraints.

Section 5: Bottom Up BIT Prioritization
This section presents a bottom-up approach to BIT prioritization. This technique attempts to rank or score the individual test's suitability for incorporation in BIT.

Section 6: Mil-Std Impact Analysis
Section 6 details results of an analysis of current Mil-Standards and Handbooks, specifically how they are affected by the allocation guidelines and optimization procedures developed in this effort.

CONTENTS

CONTENTS (Contd)

CONTENTS (Contd)

CONTENTS (Contd)

CONTENTS (Contd)

CONTENTS (Contd)

CONTENTS (Contd)

ILLUSTRATIONS

ILLUSTRATIONS (Contd)

ILLUSTRATIONS (Contd)

LIST OF FIGURES AND TABLES IN APPENDIX A

FIGURES AND TABLES IN APPENDIX A (Contd)

1.0 **INTRODUCTION**

Testability, as it applies to weapon system fault detection and diagnosis, has historically been regarded as an optional feature. Its inclusion was made only when all other "more important" functional performance issues (e.g. fly higher and faster) had been dealt with. As such, testability concerns had been addressed late in the design cycle and allocated what "leftover" resources were available. Accordingly, the maintainability of systems so developed has been less than ideal, subject to high false alarm rates and long isolation times. Weapon system complexities have been increasing with successive generations. This increase in complexity has only exacerbated this dilemna.

Recently, increasing focus has been placed on designing for testability/diagnostics throughout the development cycle. As evidence, observe that MIL-STD-2165 is intended for application throughout the design and development process. Substantial programs such as the Air Force's GIMADS and the NAVY's IDSS also serve as evidence of the trend.

We are now faced with a different challenge. Given that we consider design for testability from the earliest stages of design, how do we allocate it? In other words, given a weapon system with certain requirements for testability, how can we optimally mete out testability requirements for the constituent subsystems, and subsequent levels of indenture? The development of a process for allocating testability as a resource is, therefore, the objective of this study.

Testability allocation can be viewed as shown in Figure 1.0-1. We have relationships between testability characteristics and physical resources, system performance specification, top-level system characteristics, and system testability requirements. The goal is to cost effectively allocate the system level testability requirements and generate subsystem level requirements.

The task of developing such a methodology is substantial. A number of important related issues must be addressed. They are:

- How do we specify testability? - - What metrics do we use?
- How do we calculate the metrics that are used to specify

testability either from design data or in the field?

- How do we combine our computed/measured testability metrics from lower levels of system indenture to higher levels of indenture to demonstrate that we have met the requirements?

- Given that we can cost effectively allocate testability requirements, how do we optimally allocate BIT as a resource?

- How do we optimally modify a design to include BIT (bottom-up)?

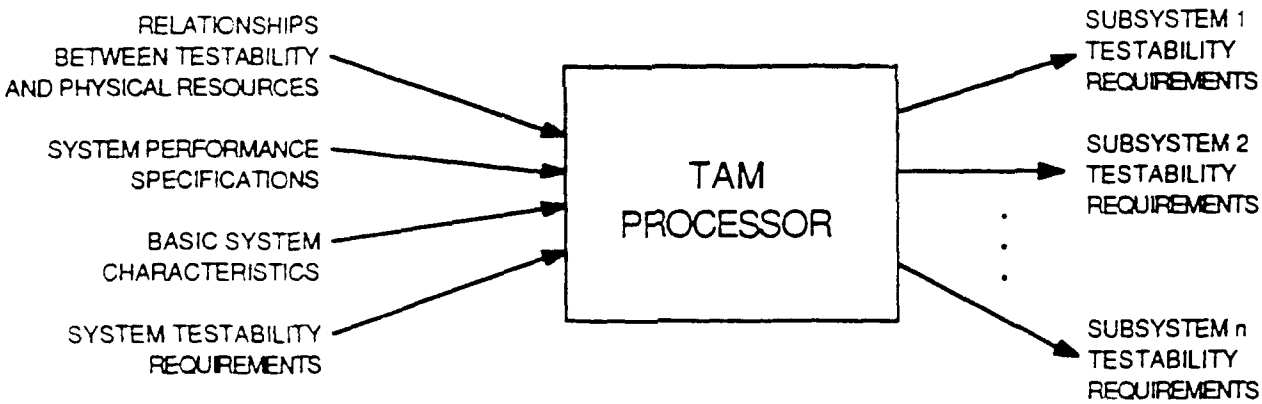


Figure 1.0-1 Testability Allocation Methodology

The objective of this study was to develop a set of approaches and algorithms that allow allocation of testability, as a resource, and answer the above mentioned questions. The overall process, methodologies, and techniques must be consistent with conventional systems engineering approaches. Further, a prototype set of software tools was developed based on these algorithms. Figure 1.0-2 depicts the overall functional architecture for the Automated Testablity Decision Tool (ATDT).

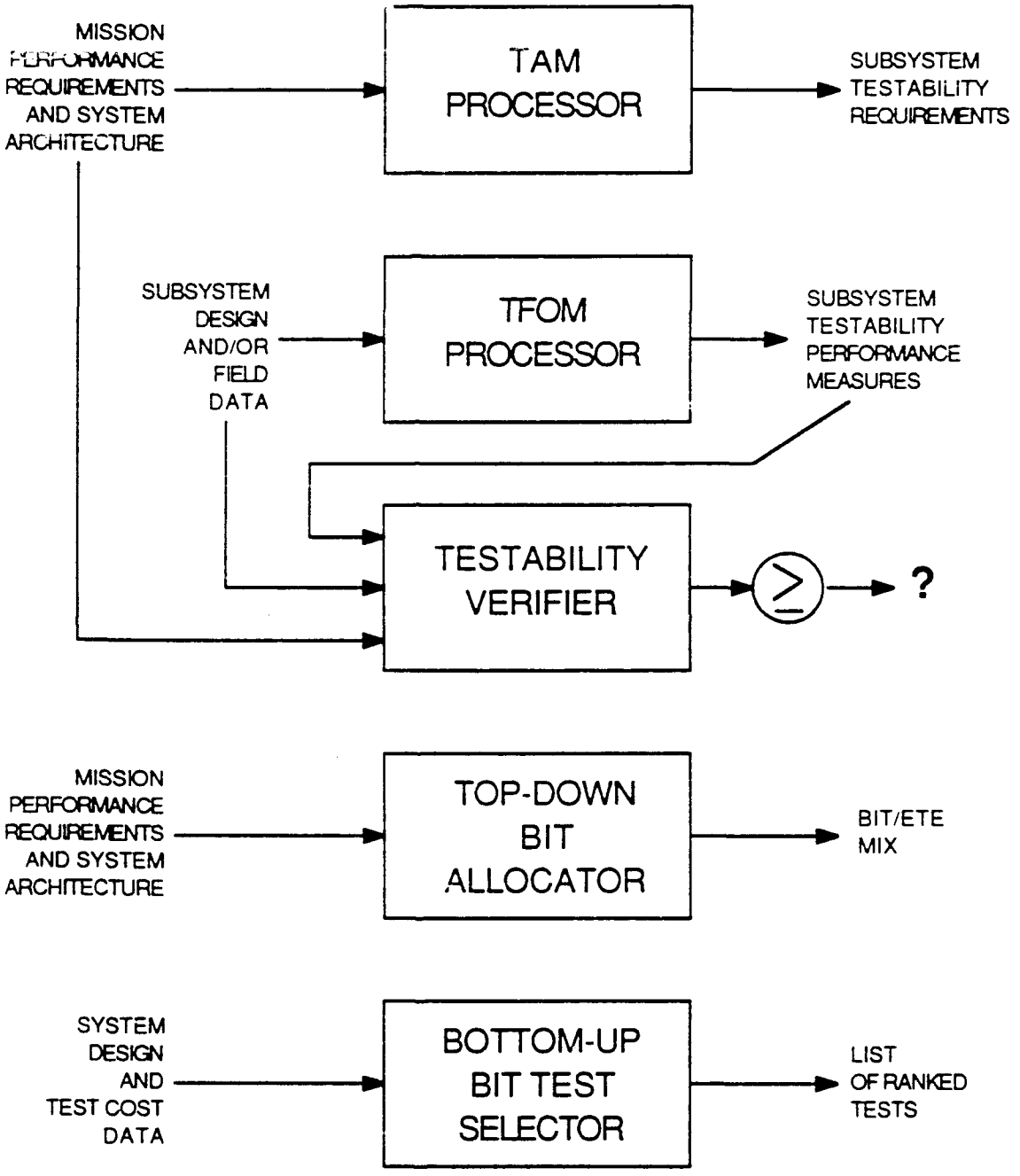


Figure 1.0-2 ATDT Functions

The approach followed in this program involved the following major tasks:

- Development of a set of Testability Figures of Merit (TFOM)
- Development of a set of Testability Allocation Methods (TAM)
- Development of a specification for integrating the TFOM with the allocation methods
- Development of a Feasibility Demonstration

1.1 Background

Testability as it appears in the literature is a "bottom-up" process. Efforts to design for testability early in a program can yield large payoffs in keeping operation and support (O & S) costs low, and reduce the mean time to repair. This, in turn, can result in a higher state of readiness and allowing higher sortie rates. Readiness is dependent on how well the user can assess the operating condition of his equipment, how quickly he can detect and locate the cause of degraded or failed components, and how efficiently he can rectify the malfunction. To be effective and reduce the high O & S costs, testability must be an integrated system development task where it is designed-in rather than added-to a weapon system.

The establishment of measurable testability allocation procedures for new prime weapon systems is the objective of this program. The requirements for testability are inextricably woven into the prime system cost effectiveness equation. The determination of "how much and where" is an integral part of the system design process. The testability process, as it applies to system design, needs to follow a top-down approach. Thus, designers must be concerned about the testability from the inception of the design process. A rigorous and trackable testability engineering procedure must be applied during every phase of the system design and engineering process: the Conceptual Design Phase, the System Design Phase, the Subsystem Design Phase and the Detailed Design Phase.

Testability, at the Conceptual Design Phase, is concerned with apportioning testability parameters, not specific design structures. System level parameters of interest at this phase are measures of system effectiveness (e. g., availability, operational readiness, mean time to repair, etc.) and overall system testability requirements (TFOMs). Allocating system testability to each major element of the system, as the same is done for reliability and availability, is desirable so that system trade-offs will include testability design requirements as parameters. Our goal is to develop a testability allocation tool available for use prior to the detailed design phase of the system. The function of this tool is to assign testability requirements to subsystems based on the overall system specifications. The input to the tool would be a description of the interrelationships between the TFOMs and the subsystems. The output would be the apportioned TFOMs for each subsystem. (See Figure 1.0-1.)

The primary task that must be completed prior to developing the tool is the definition of TFOMs and the development of meaningful relationships between TFOMs and system parameters. To facilitate the accomplishment of this task, it is necessary to characterize the mission and operational scenarios for which the development is made. It is assumed that a Statement-Of-Need (SON) exists which specifies certain mission requirements which the system designer must fulfill in order to present a viable weapon system which can assure mission success. Moreover, the operational scenario impacts the maintenance and repair philosophy which in turn imposes additional system requirements on the design engineer. In order to develop testability metrics and the requirements for the testability allocation process, the following topics are described below:

- Operational Scenario
- Maintenance Concept
- Mission/system parameters vs testability.

1.1.1 Operational Scenario

The characteristics and operational concepts of the weapon system which relate to the different levels of maintenance are described subsequently. A typical avionics mission is comprised of two time intervals (see Figure 1.1-1). They are the mission, sortie or in-flight time (T_m), and the maintenance/ checkout time (T_c). Although these time intervals may be random, it is assumed that the average time intervals are being used and

may be treated as constants. Three mission functions are fulfilled during these two intervals:

a Pre-flight
b. In-flight (or mission)
c. Post-flight

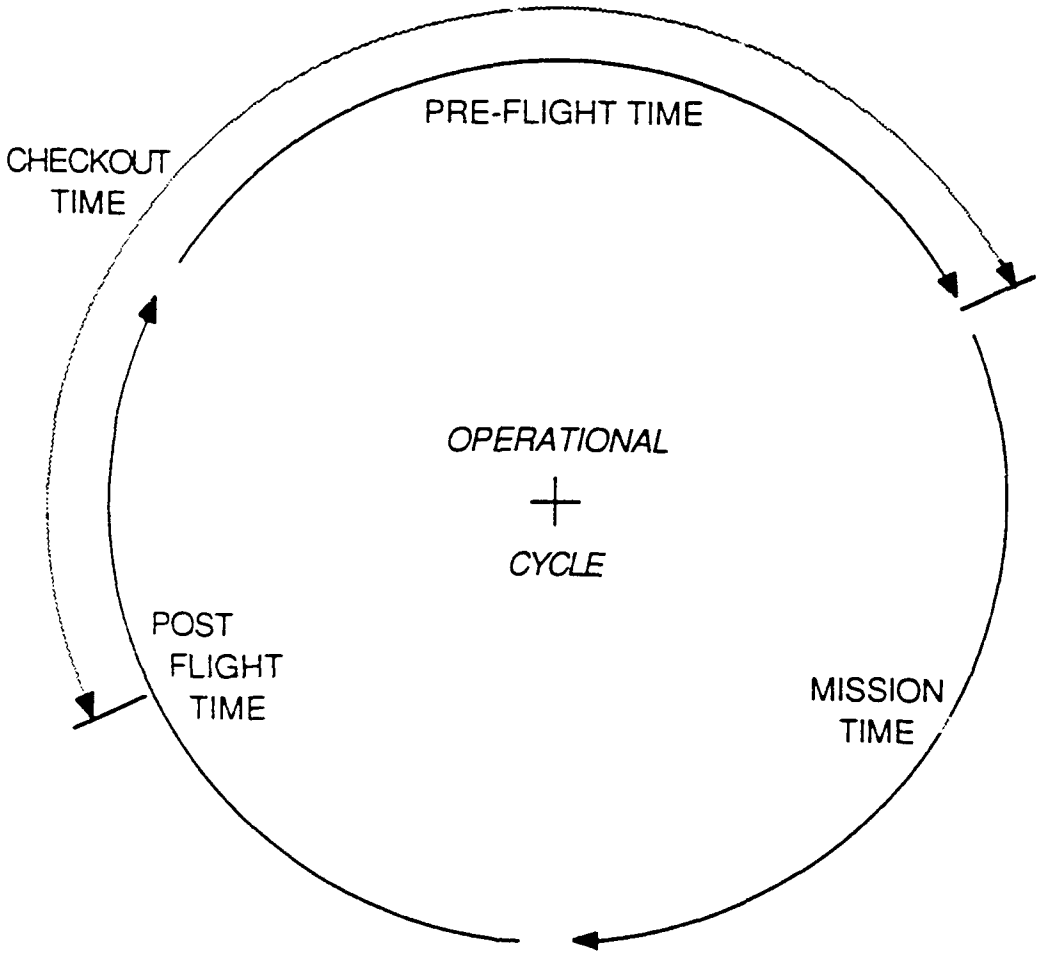


Figure 1.1-1 The Typical Avionics Mission Cycle

In the Preflight mode, high levels of fault detection (FD) are required to establish system status, but the fault isolation (FI) capability is time limited. Absolute isolation to the Line Replaceable Unit (LRU) is not necessarily required. The pre-flight functions performed are the checkout of the system and its components (using all available techniques that are pertinent) for mission readiness. If faults are detected in this interval, then repair will be instituted using the same procedures that would be used in the post-flight mode.

During the In-flight mode, isolation to the LRU level is not required, except for reconfigurable subsystems. Poor levels of FD/FI in reconfigurable subsystems will limit the achievable level of mission reliability. The in-flight functions prescribed for the system include the prime mission functions and system monitoring to assure that the weapon system continues to be mission ready. The system monitoring may involve automatic on-board diagnostics or human observation. If a mission-critical fault is detected and isolated to a redundant module while in-flight, switchover to an alternate module may be accomplished to continue the mission if this approach is part of the system design.

In the Post-flight mode, isolation is done to the LRU level, irrespective of the mission interval in which the faults are detected. Also, during the post-flight interval, repair will be instituted irrespective of the mission interval in which the faults are isolated.

The tasks comprising repair are:

Set-up:	Preparation of the Weapon System for maintenance
Isolation:	Identification of a replaceable LRU or ambiguity group of LRUs within the prime systems where the fault may reside
Rectification:	Remove and Replace the failed LRU
Checkout:	Determine whether the system has been restored and is mission ready

The times to perform the elemental repair tasks are used in all Computations of Corrective action time.

1.1.2 Maintenance Concept

The maintenance concept defines criteria governing the proposed methods of test and repair at each level of maintenance. It attempts to satisfy the performance parameters such as reliability (R), maintainability (M), and availability (A), subject to constraints, such as weight, volume and cost, associated with testability. The three levels of maintenance are:

- Organizational (O) level
- Intermediate (I) level
- Depot (D) level

The different philosophies employed at the three levels of maintenance are briefly described next.

Corrective maintenance action is initiated by reports of difficulties from the flight crew or by discovery of problems during routine post-flight inspections. The Organizational (O) level of maintenance consists of ground crews which attempt to resolve the problem on the flightline and return the aircraft to operational status. They generally have at their disposal limited-capability, Flight Line Test Equipment (FLTE). The function of the O-level crew is to

- isolate the problem to the level of a suspect LRU,
- remove and replace the LRU,
- test the system.

O-level personnel are specifically forbidden from disassembling an LRU and attempting to repair it. The replaceable unit diagnosed as faulty by the O-level maintenance crew is then removed and sent to the second level of maintenance, generally the intermediate (I) level for repair, or is discarded. In future systems the I-level will be eliminated and the faulty LRUs will be sent to the D level.

The intermediate (I) level shops have at their disposal more sophisticated and accurate test devices, tools, and automatic test equipment. Thus, more extensive fault diagnosis is possible at the I-level. This level of maintenance is permitted to

- access the interior of the LRU,
- remove and replace any shop replaceable unit (SRU) suspected or proven to be faulty.

Normally, I-level does not repair complex SRUs, e. g., a transistor on a circuit card. If an SRU is of sufficient value to warrant its repair, it will be sent to the next level of maintenance, the Depot (D) level.

At the D-level, repair is effected either by the original manufacturer or at a military depot remote from the airfield.

1.1.3 Mission/System Parameters Vs Testability

Definitions for operational readiness, mission time, mission reliability, critical failure detectability and system maintainability provide relationships between the mission scenario and system design criteria in which testability plays a part. Included in these relationships are criteria such as failure rates and mean time to repair (MTTR). Using these interrelationships as tied to life cycle costs, optimization techniques can be applied to allocate testability into the system design and across levels of indenture.

Operational Readiness (P_{or}) is the probability that the weapon system is ready to start operating when the next mission is scheduled to commence.

Mission time and checkout time (T_m and T_c respectively) have already been defined and discussed to some extent. They are assumed to be constants in the developments that follow.

Mission Reliability ($R(T_m)$) is the probability that no mission critical failure(s) occur during the mission time. A useful equivalent is given by $Q(T_m) = 1 - R(T_m)$, the probability that at least one mission critical failure occurs during the mission time.

Detectability (D or Pr(D)) is the probability that a mission critical failure that has occurred will be detected during the mission or during post-flight checkout by human observation, Built-in Test (BIT), External Test Equipment (ETE) including FLTE or any other means.

Maintainability The measure of the ability of an item to be retained in, or restored to, specified condition when maintenance is performed during the course of a specified mission profile.

Many other parameters are of interest in system analysis and design as far as testability is concerned. These other parameters will be defined as necessary in this report within the framework of the TFOM/TAM integration.

1.1.4 Summary

The initial steps in the development of an Automated Testability Decision Tool, have been stated in prior subsections. Discussion has been limited to elements of the:

- background in terms of operational scenario and maintenance concept
- relationships between mission parameters and testability.

The diagnostic systems considered in this study are provided by:

- direct human observation of the equipment in-flight and on the flight line;
- BIT exercized in-flight during normal mission operation;
- Flight line test equipment (FLTE) in the form of external test equipment (ETE) which may be automatic and may be used in conjunction with BIT.

The function of these diagnostic systems is to provide rapid detection and isolation of faults so that the weapon system can be returned to an operational condition as quickly as possible.
The development and selection of the TFOMs described in section 2.0 are such that:

- they reflect the objective of the diagnostic systems stated

previously;

- they are measurable during system design, acceptance and deployment;
- they are relatable to prime support performance parameters of reliability, maintainability, availability, and cost.

The study is also driven by the following requirements for testability allocation.

- Testability allocations should be applied top-down; that is, apportion system requirements to subsystems at the system level first; followed by the subsystem to the LRU level. Care should be taken not to specify diagnostic requirements blindly to lower levels unless they fully satisfy the system-level requirements.

- The 2-level (O and D) maintenance philosophy together with the operational scenario imposes the requirement that the isolation to the LRU level must be unambiguous. Thus, high levels of diagnostics within subsystems must be specified.

- The mission functions impose additional requirements on the testability allocation tool since one needs to detect and isolate differently for different modes of operation. For example, inflight testability of a reconfigurable system should be allocated to meet mission reliability requirements.

1.2 Report Organization

The ATDT report organization follows:

2.0 TFOM DEVELOPMENT

This section defines and computes the testability requirements as measured by the Testability Figures of Merit (TFOMs) in precise, calculable and measurable engineering terms.

3.0 TAM DEVELOPMENT

This section defines and develops a generalized procedure for allocating System testability requirements through lower levels of indenture to the Line Replaceable Unit (LRU).

4.0 TFOM/TAM INTEGRATION

This section details the complex interactions between diagnostic system performance parameters as measured by the TFOMs and system reliability (R), availability (A), maintainability and Life Cycle Cost (LCC). These relationships are used to develop the procedure to integrate the cost functions (objective and constraints) and the allocation methodology.

A top-down approach to BIT prioritization has concentrated on interrelating design and mission parameters that involve BIT in the system design. This approach reduces to a problem of allocating BIT used in the system design. Thus, the top-down approach to BIT prioritization is a subset to the Testability Allocation Method, and is treated in the TFOM/TAM integration section.

5.0 BOTTOM-UP BIT PRIORITIZATION

This Section presents a bottom-up approach to BIT prioritization. This technique attempts to rank or score the individual test's suitability for incorporation in BIT, in line with the objectives of RADC's CAD-BIT program.

6.0 MIL-STD IMPACT ANALYSIS

This section is the first part of the Development of a Feasibility Demonstration task. The second part is comprised of the ATDT Software prototype. The objective of the MIL-STD impact analysis is the identification of those military standards, specifications and handbooks that would be affected by the results of ATDT. There are two ways that these documents can be impacted, and these can be described as a "producer/consumer" relationships. As "producer" or data sources, the MIL-STD's, MIL-SPEC's and Handbooks do not provide all the data required by ATDT. As data "consumers",

these documents may be able to make use of the ATDT outputs.

7.0 REFERENCES

This section presents references and bibliography used in the development of ATDT.

APPENDIX A

This appendix includes figures and tables taken from MATE Guide (G3V3P2; Appendix E and section 7) dated 1 April 1985.

2.0 TFOM DEVELOPMENT

2.1 Introduction.

The objective of the first task in providing an Automated Testability Decision Tool is to define, compute and assess the testability metrics (TFOMs) as applied to a weapon system, in precise and measurable engineering terms. The quality of the testability features included in the system design should be measurable by the TFOMs. In addition to providing metrics for the testability in a system, the TFOMs must be translatable to the specification of system design requirements as well as system performance specifications. This requires that a choice of TFOMs be made which are geared specifically to these purposes.

The goal of measuring the quality of a system's testability may be decomposed as follows:

- The set of TFOM's must be complete in that it characterizes all facets of a system's testability.

- The individual TFOM's must exhibit a minimum amount of redundancy. That is to say, each figure of merit should stand alone from the others. No specific TFOM should characterize a testability attribute that is also characterized, in one form or another, by another TFOM. For example, one TFOM, a cumulative distribution representing the likelihood of fault isolation of differing ambiguity group sizes (Fault Isolation Resolution), characterizes testability attributes that are also characterized by another TFOM, the fraction of faults isolated (FFI). As such, only one of these two may be used.

- The figures of merit should be meaningful to different observers. These range from the planner who thinks in terms of missions, functions, and mission effectiveness, to the design engineer who relates to terms such as components and reliabilities.

- The TFOM's must be translatable across arbitrary levels of indenture. Calculation procedures must be available to combine TFOM's at a given level of indenture to form the TFOM's for the next higher level of indenture (e.g., component level TFOM's to SRU-level TFOM's, SRU-level TFOM's to LRU-level TFOM's , etc.).

The set of TFOM's to be identified were to be usable both to characterize the testability of existing systems as well as to specify the required level of testability for systems under development. This dual role places two additional requirements on our set of TFOM's.

- The testability figures of merit must be calculable from design data, at varying stages of design (e.g., PDR and CDR).

- The set of TFOM's, as testability specifiers, must not fully constrain the testability design of any given system. If they were to fully constrain the design, they would in effect constitute a design -- not a specification; specifications attempt to characterize performance, not designs. Opportunities for innovation should be allowed and encouraged.

2.2 Organization and Approach

The section organization and corresponding approach taken in the development of TFOMs follows:

2.2.1 TFOM Survey (existing and postulated)

2.2.2 Top-Down Analysis of Testability Requirements

This step involves a sensitivity analysis of the system performance models (availability, operational readiness, life-cycle-cost) to

determine which testability attributes, such as *false pull rate* or *expected time to fault isolation*, were needed.

2.2.3 Selection and Refinement of TFOM's
This step provides detailed descriptions of the algorithms for computation of the selected TFOMs.

2.2.4 TFOM Translation Analysis
This step analyzes the selected TFOM's for ease of use, and the ease of translation to design implementation across levels of system indenture. The equations derived combine TFOM's at a given level of indenture to form those TFOM's that characterize the testability at the next higher level of indenture.

2.2.1 TFOM Survey.
The survey of testability figures of merit had several objectives. The first was to simply identify all TFOM's used or postulated by the military, industry, and academia. The second objective was to understand the computational requirements and value of each of the TFOM's identified. The third objective was to categorize the TFOM's for evaluation in subsequent steps of the investigation.

The remainder of this subsection is organized under three topics. The first is a discussion of the approach taken for the survey. The second topic addresses the findings of the survey. Finally, a commentary on the findings is presented.

2.2.1.1 Survey Approach.
The TFOM survey was carried out in three phases: a literature survey, a literature review and analysis, and a categorization of TFOM's discovered.

2.2.1.1.1 Literature Survey.
The goal of the literature survey was to identify and acquire references that described testability figures of merit. These TFOM's may have been used as military or industry specifiers for testability, by existing tools

and methods for measurement of the goodness of testability, or postulated for use as a specifier or metric.

The literature survey was initiated with a keyword search in two data bases: NTIS and INSPEC. The keywords used are given in Figure 2.2-1. Aside from the keywords, the names of several well published authors were used in the search. They were: W. R. Simpson for his work in the development of a highly respected testability analysis tool, STAMP™, K. Pattipati for his work in the development of the "TEST" algorithm for fault tree formulations, and J. Bussert for his work in evaluating many of the existing testability tools. In all, there were 93 hits against those keywords and authors. Abstracts for the 93 were acquired and reviewed to determine which technical papers, reports, and books to obtain.

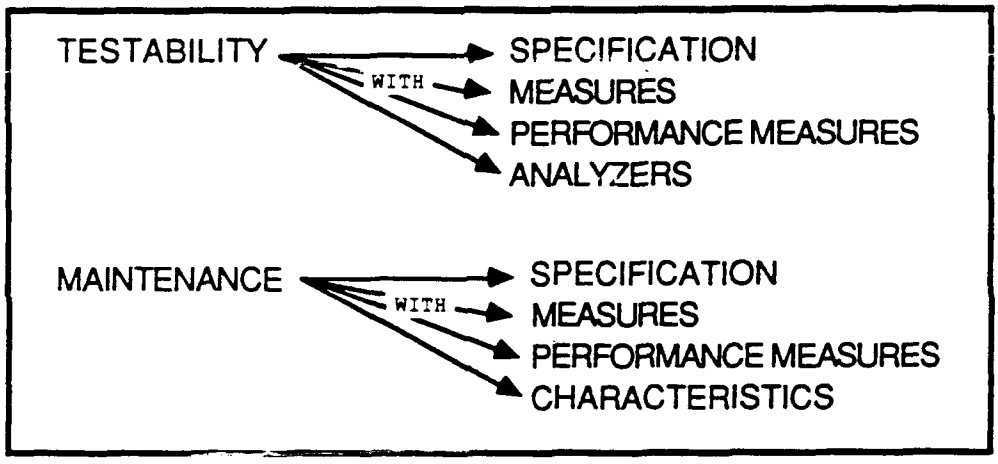


Figure 2.2-1 Keyword Strategy Used for Literature Search

In addition to the database searches, Harris GSSD has a substantial collection of papers, reports, and books as sources for other testability-related efforts. These include papers and reports describing the IDSS-related Weapon System Testability Analyzer (WSTA). They were reviewed to determine their relevancy to this investigation. Those that were deemed to be relevant were used. Further, the reference lists from those publications as well as from papers acquired through the keyword search were used to identify additional publications. In the end, more than 80 publications were acquired for in-depth review.

2.2.1.1.2 Literature Review and Analysis.

The papers that were identified and acquired during the literature survey were reviewed with the focus of identifying descriptions of testability figures of merit. The TFOM's identified were further analyzed to determine their computational requirements and the testability facet(s) that they characterized. Ultimately some 100 TFOM's were identified and evaluated. The details of the findings are reported in Section 2.2.1.2.

2.2.1.1.3 Categorization of TFOM's.

The purpose of TFOM categorization was to organize the various figures of merit for evaluation. Each TFOM was reviewed and categorized in two ways. First, they were analyzed to determine their applicability to models of system performance (i.e. availability, mission reliability, and life-cycle cost models). This evaluation did not specifically examine the models, rather it subjectively examined the potential use of each TFOM as an independent variable in such models.

The second categorization of each TFOM was according to the particular facets of testability that it characterized. Specifically, testability has two roles, that of detection and that of isolation. Each of those roles have performance facets. This categorization subjectively analyzed each TFOM in terms of its ability to characterize those facets.

These categorizations were used as filters to reduce the set of TFOM's to a manageable size. That subsequent set would then be further analyzed and pruned to a desired set. The details of these categorizations are given in the following section.

2.2.1.2 TFOM Survey Findings.

The TFOM survey findings began with a summary of the RADC report *BIT/External Test Figures of Merit and Demonstration Techniques* by Pliska et al. (see bibliography). It had a similar goal, to identify a complete set of assessable testability figures of merit. That document was published in 1979 and thus predated most of the currently available computer-based tools for testability analysis. As such, the TFOM's

reported therein were derived from the then current military publications. In the following section, 2.2.1.2.1, a summary of the TFOM's identified in that RADC report is given. Following the summary, a review of the additional figures of merit discovered in our study is given in Section 2.2.1.2.2.

2.2.1.2.1 Summary of 1979 RADC TFOM Report.

RADC Report No. RADC-TR-79-309, entitled *BIT/External Test, Figures of Merit and Demonstration Techniques* by Pliska et al., identified and evaluated TFOM's for measuring testability (as provided by built-in test and external test equipment) performance and incremental penalties (e.g., weight, volume, power) characteristics.

In all, the RADC report defined 18 figures of merit. They were:

FFD -	Fraction of Faults Detected
FFA -	Fraction of False Alarms
FFSI -	Fraction of False Status Indications
T_{FD} -	Mean Fault Detection Time
T_B -	Mean Time Required for BIT/ETE Executions
F_B -	Frequency of BIT/ETE Executions
TT -	Test Thoroughness
FIR(I) -	Fault Isolation Resolution
FFI -	Fraction of Faults Isolated
T_{FI} -	Mean Fault Isolation Time
MPSL -	Maintenance Personnel Skill Level
$MTBF_{B/E}$ -	Mean Time Between Failures in BIT/ETE
$MTTR_{B/E}$ -	Mean Time To Repair Faults in BIT/ETE
$A_{B/E}$ -	BIT/ETE Availability
MTTR -	Mean Time To Repair
A -	System Availability
FFP -	Fraction of False Pulls
FEFI -	Fraction of Erroneous Fault Isolations

Note that these TFOM's and the definitions which follow (which were extracted from the 1979 RADC Document) occasionally use different nomenclature than is used in later sections of this report.

Each of these above TFOM's is reviewed in the following paragraphs.

FFD - Fraction of Faults Detected

Two distinct definitions for this figure of merit were reported. The first, FFD_A is defined as the fraction of **all** faults detected (or detectable) by BIT or ETE.

$$FFD_A = Q_{BDF}/Q_F \quad (2\text{-}1)$$

where Q_{BDF} is the quantity of faults detected by BIT or external test equipment, and Q_F is the quantity of all faults.

The second variant of this TFOM is FFD_D. It is defined as the fraction of **detected** faults detected (or detectable) using BIT or ETE. In other words, FFD_D is the ratio of BIT/ETE detected faults to those faults that are detected (or detectable) by any means.

$$FFD_D = Q_{BDF}/Q_{FD} \quad (2\text{-}2)$$

where Q_{FD} is the quantity of faults detectable by any means.

In both definitions, Q_{BDF} , Q_F , and Q_{FD} exclude the occurrence of false alarms. The difference between these metrics is debatable and is a function of the denominators. The apparent difference between Q_F and Q_{FD} is the number of faults that occur and are never detected, say $(Q_F - Q_{FD})$. It can be argued that any fault that is never detected by degradation in mission function is not a fault relative to the mission requirements. As a consequence, one would expect that the quantity $(Q_F - Q_{FD})$ should approach zero leaving one form for FFD.

FFD can be analyzed using predicted failure rates. It can be verified through proper field data collection.

FFA - Fraction of False Alarms

FFA is defined as the fraction of false alarms caused by BIT or ETE. False alarms are considered to be any indicated faults due to faulty BIT or ETE, out-of-tolerance conditions, or transient conditions. Real but intermittent

faults are not classified as false alarms. FFA is mathematically defined by the following quotient:

$$FFA = Q_{FA}/Q_{BIF} \quad (2\text{-}3)$$

where Q_{FA} is the quantity of BIT or ETE false alarms, and Q_{BIF} is the quantity of indicated faults due to BIT or ETE.

FFA can be analyzed using predicted failure rates and verified through proper field data collection techniques.

FFSI - Fraction of False Status Indications

FFSI is defined as the fraction of false status indications, including both false alarms and missed detections, that are caused by BIT or ETE. This ratio takes into account all failures, both detected and undetected.

$$FFSI = (Q_{FA} + Q_{UD}) / (Q_{BIF} + Q_{UD}) \quad (2\text{-}4)$$

where Q_{FA} is the quantity of BIT or ETE false alarms, Q_{UD} is the quantity of undetected faults, and Q_{BIF} is the quantity of failure reports due to BIT or ETE.

FFSI can be analyzed during design using predicted failure rates and verified through proper field data collection techniques.

T_{FD} - Mean Fault Detection Time

T_{FD} is defined as the average latency period between the occurrence of a fault to the point in time required for BIT or ETE to report its existence. Mathematically, T_{FD} is formulated as follows:

$$T_{FD} = \left(\sum_{i=1}^{Q_{BDF}} t_i \right) / Q_{BDF} \quad (2\text{-}5)$$

where t_i is the time required to detect the i^{th} BIT/ETE detectable fault, and Q_{BDF} is the quantity of BIT/ETE detectable faults.

This TFOM can be analyzed using techniques similar to those prescribed by MIL-HDBK-472, procedure 2, or RADC-TR-78-169. These compute the weighted average of times derived through time line analysis. *In the field,* T_{FD} can be verified through direct time measurements.

T_B - Mean Time Required for BIT/ETE Executions

T_B is the average time required to execute a BIT or ETE routine. Mathematically, T_B is formulated as follows:

$$T_B = \left(\sum_{i=1}^{N_B} t_{Bi} \right) / N_B \qquad (2\text{-}6)$$

where t_{Bi} is the execution time required for the i^{th} BIT/ETE test, and N_B is the quantity of BIT/ETE tests under consideration.

T_B can be analyzed using straight forward time line analysis. In the field, T_B can be verified through direct time measurements.

F_B - Frequency of BIT/ETE Executions

F_B is defined as the frequency of occurrence of cyclic BIT or ETE tests. Mathematically, F_B is formulated as follows:

$$F_B = (T_{TBE} + T_I)^{-1} \qquad (2\text{-}7)$$

where T_{TBE} is the time required for the complete execution of BIT and/or ETE routines, and T_I is the idle time between cycles.

F_B can be analyzed using time line analysis and verified through direct time measurements.

TT - Test Thoroughness

TT is the fraction of an equipment/system that is tested by BIT or ETE relative to the entire equipment/system.

$$TT = C_{BE} / C_T \tag{2-8}$$

where C_{BE} is the quantity of components or functions, possibly weighted by failure rates, that are tested by BIT and/or ETE, and C_T is the total number of components or functions that comprise the system.

TT can be analyzed during design using failure rate predictions and verified in the field through proper data collection techniques.

FIR(L) - Fault Isolation Resolution

FIR(L) is the cumulative probability that any detected fault can be isolated by BIT or ETE to an ambiguity group of size L or less.

$$FIR(L) = Q_{IL}/Q_{FD} \tag{2-9}$$

where Q_{IL} is the quantity of detected faults that may be isolated to L or fewer replaceable units, and Q_{FD} is the quantity of detectable faults.

FIR(L) can be analyzed during design using predicted failure rates and verified through proper field data collection techniques.

FFI - Fraction of Faults Isolated

FFI is defined as the fraction of faults isolated to some level specified by the maintenance concept. Its mathematical formulation is:

$$FFI = Q_{IB}/Q_{FD} \tag{2-10}$$

where Q_{IB} is the quantity of detected faults that may be isolated to the level specified by the maintenance concept, and Q_{FD} is the quantity of detectable faults. Note that this definition is dependent on the specified level of ambiguity and, as such, is subject to interpretation.

FFI can be analyzed during design using predicted failure rates and verified through proper field data collection techniques.

T_{FI} - Mean Fault Isolation Time

T_{FI} is defined as the average time required to isolate a fault using BIT or ETE. Mathematically, T_{FI} is formulated as follows:

$$T_{FI} = \left(\sum_{i=1}^{Q_{BDF}} t_{FIi} \right) / Q_{BDF} \qquad (2\text{-}11)$$

where t_{FIi} is the time required to isolate the i^{th} fault using BIT or ETE, and Q_{BDF} is the quantity of BIT/ETE detectable faults.

This TFOM can be analyzed using a failure rate weighted average of times derived through time line analysis. In the field, T_{FI} can be verified through proper data collection.

MPSL - Maintenance Personnel Skill Level

MPSL is user defined measure of the skill required to perform a certain task, set of procedures, or other maintenance related activity. The resulting score has an arbitrary range and is highly subjective. Because of its subjectivity, MPSL would not appear to be reliable.

MPSL can be analyzed by evaluating maintenance task skill requirements. In the field, MPSL can be verified through proper data collection.

$MTBF_{B/E}$ - Mean Time Between Failures in BIT/ETE

$MTBF_{B/E}$ is defined as the average time period between the occurrence of faults within the BIT or ETE subsystems. Observe that those failures may result in either false alarms or missed detections. Mathematically, $MTBF_{B/E}$ is formulated as follows:

$$MTBF_{B/E} = \left(\sum_{i=1}^{N_{B/E}} \lambda_{B/Ei} \right)^{-1} \qquad (2\text{-}12)$$

where $\lambda_{B/Ei}$ is the failure rate of the i^{th} BIT or ETE component, and $N_{B/E}$ is the number of BIT or ETE components.

$MTBF_{B/E}$ can be analyzed using techniques similar to those prescribed by MIL-HDBK-217. In the field, $MTBF_{B/E}$ can be verified through proper data collection.

$MTTR_{B/E}$ - Mean Time To Repair Faults in BIT/ETE

$MTTR_{B/E}$ is defined as the average time required to repair a fault in a BIT or ETE subsystem. $MTTR_{B/E}$ takes into account times for isolation and rectification (i.e., the component replacements and subsequent checkout). Mathematically, $MTTR_{B/E}$ is formulated as follows:

$$MTTR_{B/E} = \left(\sum_{i=1}^{N_{B/E}} \lambda_{B/Ei}\, t_{RBEi} \right) / \left(\sum_{i=1}^{N_{B/E}} \lambda_{B/Ei} \right) \qquad (2\text{-}13)$$

where $\lambda_{B/Ei}$ is the failure rate of the i^{th} BIT or ETE component, t_{RBEi} is the time required to isolate and rectify a failure in the i^{th} BIT or ETE component, and $N_{B/E}$ is the number of BIT or ETE components.

This TFOM can be analyzed using techniques similar to those prescribed by MIL-HDBK-472 or RADC-TR-78-169. In the field, $MTTR_{B/E}$ can be verified through proper data collection or by using techniques in MIL-STD-471.

$A_{B/E}$ - BIT/ETE Availability

$A_{B/E}$ is defined as the probability that BIT or ETE will be operational at some arbitrary time t_1, given that it is operational at some previous time t_0. Its mathematical formulation is:

$$A_{B/E} = MTBF_{B/E} / (MTBF_{B/E} + MTTR_{B/E}) \qquad (2\text{-}14)$$

where both $MTBF_{B/E}$ and $MTTR_{B/E}$ are TFOM's described previously in this section.

MTTR - System Maintainability

MTTR is defined as the average time required for repairing all system/equipment faults. The impact of BIT or ETE is seen in the isolation component of this measure. Mathematically, MTTR is formulated as follows:

$$MTTR = \left(\sum_{i=1}^{N} \lambda_i t_i \right) / \left(\sum_{i=1}^{N} \lambda_i \right) \qquad (2\text{-}15)$$

where t_i is the time required to isolate and rectify a fault in the i^{th} system component, λ_i is the predicted failure rate of the i^{th} system component, and N is the number of system components.

This TFOM can be analyzed using techniques similar to those prescribed by MIL-HDBK-472 or RADC-TR-78-169. In the field, MTTR can be verified through proper data collection or by using techniques in MIL-STD-471.

A - System Availability

A is defined as the probability that a system will be operational at some arbitrary time t_1, given that it is operational at some previous time t_0. Its mathematical formulation is:

$$A = MTBF / (MTBF + MTTR) \qquad (2\text{-}16)$$

where MTTR is a TFOM described previously in this section, and MTBF is defined as:

$$MTBF = \left(\sum_{i=1}^{N} \lambda_i \right)^{-1} \qquad (2\text{-}17)$$

λ_i is the predicted failure rate of the i^{th} system component.

MTTR can be analyzed during design using techniques from MIL-HDBK-217 and MIL-HDBK-472. It can be verified in the field using proper data collection techniques, or methods from MIL-STD-781 and MIL-STD-471.

FFP - Fraction of False Pulls

FFP is defined as the fraction of replaceable units within a system that are unnecessarily replaced during maintenance. FFP is mathematically defined by the following quotient:

$$FFP = Q_{GRI}/Q_{RR} \quad (2\text{-}18)$$

where Q_{GRI} is the quantity of good replaceable units that are removed, and Q_{RR} is the quantity of replaceable units that are removed during maintenance, both good and bad.

FFP was determined to be both indirectly analyzable and verifiable. This is because it may be derived from another TFOM, specifically FIR(L). FFP is then determined by computing the expected ambiguity group size and allowing for some predetermined replacement strategy.

FEFI - Fraction of Erroneous Fault Isolations

FEFI is defined as the fraction of incorrect fault isolation results. FEFI is mathematically defined by the following quotient:

$$FEFI = Q_{EFIR}/Q_{FIR} \quad (2\text{-}19)$$

where Q_{EFIR} is the quantity of isolation results that are incorrect, and Q_{FIR} is the total quantity of isolations, both good and bad.

FEFI was determined not to be analyzable during design, but can be computed in the field using appropriate data collection techniques.

Commentary on TFOM's Reported

The results of the RADC report are noteworthy. They are especially important, in that potential military TFOM's were systematically identified and analyzed. Each of the TFOM's that were identified, characterized one or more facet of testability (BIT/ETE functions). That characterization is depicted in Figure 2.2-2.

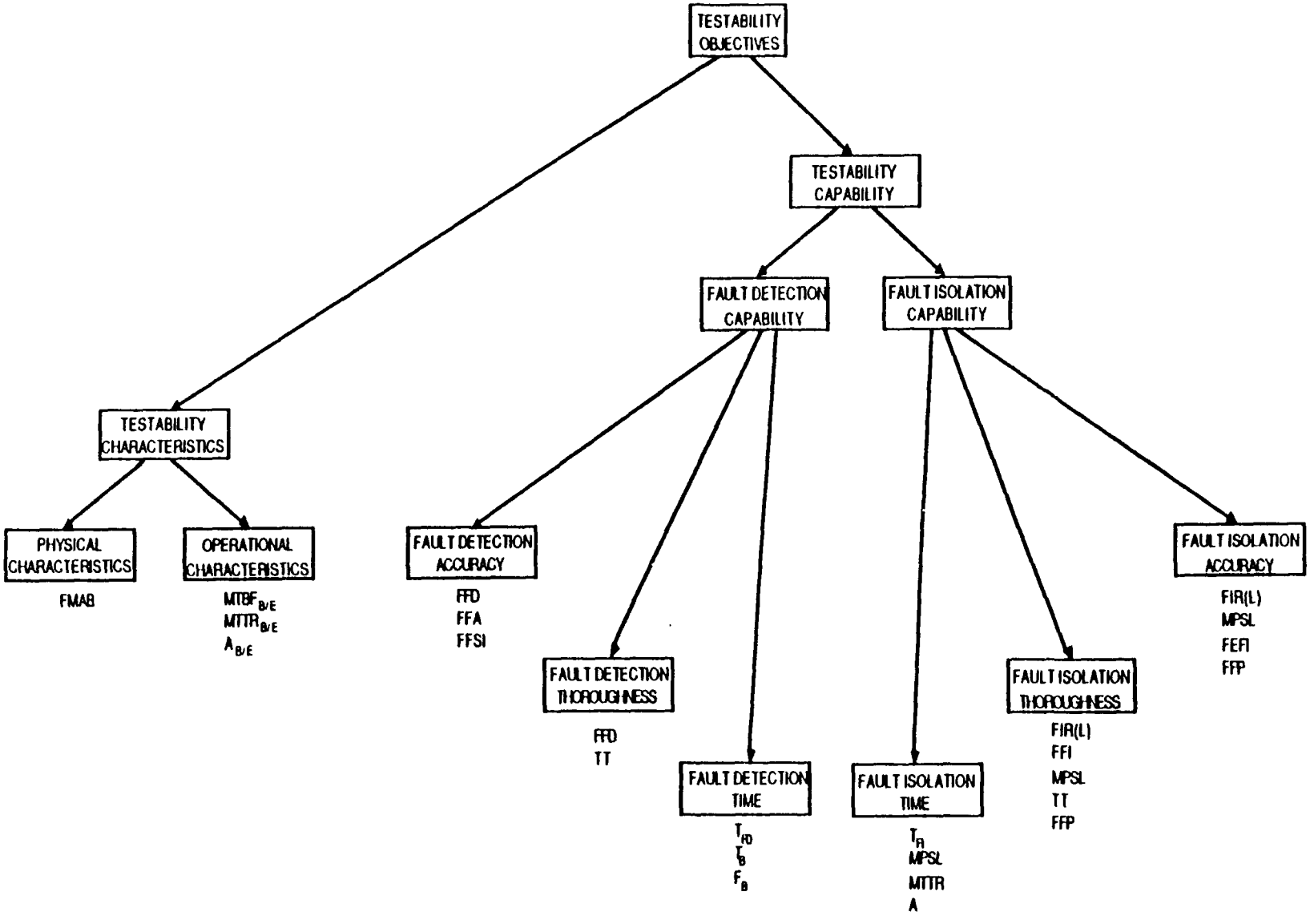


Figure 2.2-2 Categorization of TFOM's Identified in RADC Report RADC-TR-79-309

The RADC report did have some shortcomings. First, some sweeping assumptions were made in an effort to qualify the TFOM's as analyzable and verifiable. The first of these assumptions was that the number of detected faults would be the same as the number of detectable faults. The latter is an upper bound of the former, but they are not equivalent.

A second potentially dangerous assumption lies in the repeated assessment that various TFOM's would be verifiable using field data collection techniques. As reported in RADC Report RADC-TR-85-268 by Simpson et al., data collection techniques to assess some of the TFOM's identified would require a substantial change to the current Air Force maintenance reporting system.

Finally, no attempt was made to assess the degree to which the TFOM's characterize testability. For example, FFD was determined to be a measure of fault detection thoroughness. The question still remains, does it completely characterize thoroughness? As a result we do not know exactly where the set of 18 TFOM's are deficient, overlapping or exactly sufficient.

2.2.1.2.2 Additional TFOM's.

With the RADC report by Pliska et al. as a baseline, additional publications were reviewed to identify other figures of merit. In all, 100 were identified. Those are presented in the following section as a listing. The format of that listing is as follows. The mnemonics typically used for the TFOM's are provided along with brief definitions and, in some cases, calculative details. The definitions are organized by their sources (e.g. a particular tool or military report). In cases where a particular TFOM was uncovered from more than one source (e.g. the identical TFOM calculated by both the WSTA and LOGMOD tools), it is not repeated. Finally, each TFOM is assigned to one of three categories: *GENERAL*, *MODEL*, or *ADVISORY*.

- [*Advisory*] TFOM's are those deemed to be measures of some specific aspect of testability. They may be either abstract in nature or represent concrete design parameters.

- [*Model*] TFOM's are those deemed to be concrete design characteristics that can play roles in mathematical models for availability and life-cycle costs.

- [*General*] TFOM's are those that are abstract measures or grades of testability. Their primary use is to gauge testability goodness. Typically, they have no roles in any mathematical models.

TFOM's IDENTIFIED IN MILITARY REPORTS AND DOCUMENTS

FFD Fraction of Faults Detected (effectively the probability that a fault will be detected given that one occurs)
[MODEL]

FFA Fraction of False Alarms (effectively the probability that any given maintenance action is a result of a false alarm ,i.e. a report of a fault when none exists)
[MODEL]

FFSI Fraction of False Status Indications (effectively the probability that a given status indication,good or bad, is erroneous)
[MODEL]

T_{FD} Mean fault detection time, or the expected latency between the occurrence of a fault and its subsequent detection
[MODEL]

T_B Mean time required for BIT/ETE executions
[MODEL]

F_B Frequency of BIT/ETE executions
[MODEL]

TT Test Thoroughness (the fraction of equipment components/subsystems that are tested)
[GENERAL]

FIR(L) Fault Isolation Resolution (this is a cumulative probability distribution that measures the likelihood that a fault will be isolated to an ambiguity group not greater in size than L)
[MODEL]

FFI Fraction of Faults Isolated (effectively the probability that any fault will be isolated). This measure is often taken to be equivalent to FIR(1). Another interpretation has it equivalent to FIR(n) for arbitrary value of n.

[MODEL]

T_{FI} Mean fault isolation time

[MODEL]

MPSL Maintenance Personnel Skill Level (a numeric characterization of the ability of maintenance personnel). This measure is highly subjective and has no units or range.

[GENERAL]

$MTBF_{B/E}$ Mean Time Between Failures in BIT/ETE systems

[MODEL]

$MTTR_{B/E}$ Mean Time To Repair faults in BIT/ETE systems

[MODEL]

$A_{B/E}$ BIT/ETE availability (effectively the probability that BIT/ETE will be operable in an error-free fashion at any arbitrary point in time)

[MODEL]

MTTR Mean Time To Repair a fault in a system

[MODEL]

A System Availability (effectively the probability that a system will be available for a mission at any given time)

[MODEL]

FFP Fraction of False Pulls (effectively the probability that a given component/subsystem, when replaced during corrective maintenance, was done so unnecessarily)

[MODEL]

FEFI Fraction of Erroneous Fault Isolation results (effectively the probability that any given fault isolation conclusion is incorrect)

[MODEL]

FMAB Fraction of Memory Allocated to BIT (that portion of a system/subsystem/component's memory that is dedicated to BIT)

[ADVISORY]

ITFOM Inherent Testability Figure of Merit (a checklist-derived value that attempts to measure the goodness of a system's testability). There are various versions of this class of TFOM.

[GENERAL]

CND_b Cannot Duplicate Events Burden (the percentage of all maintenance actions that are classified as cannot duplicate events)
[MODEL]

CND_r Cannot Duplicate Events Rate (the time rate at which cannot duplicate events occur)
[MODEL]

FAR False Alarm Rate (the time rate at which false alarms occur)
[MODEL]

FMAI Fraction of Maintenance Actions that result in an Isolation
[MODEL]

NDR Non-Detection Rate (the time rate at which failures occur that are not detected)
[MODEL]

NIR Non-Isolation Rate (the time rate at which failures occur that are not isolated)
[MODEL]

TFOM's IDENTIFIED IN IDSS WSTA

L_I Inherent fault isolation level (In its weighted form, this measure is effectively the probability that a given fault resides in an ambiguity group of size i. Note that this is a distribution.)
[MODEL]

I_K Component involvement ratio (The percentage of ambiguity groups in which the kth component is a member. Note that this TFOM is a vector.)
[ADVISORY]

AFP_j Aggregate Failure Probability of ambiguity group j. (Effectively the probability that the j^{th} ambiguity group will fail. Note that this TFOM is a vector.)
[MODEL]

EAG Expected Ambiguity Group size
[MODEL]

AG_{MAX} Size of the largest Ambiguity Group
[GENERAL]

AG_{MIN} Size of the smallest Ambiguity Group
[GENERAL]

FIG_i Fault Isolation Gain of the i^{th} test (The improvement made if feedback loops, in which test i is nested, are broken at test i. Note that this TFOM is a vector.)

[ADVISORY]

FBC_q Feedback Loop Composition (The set of tests and components that make up the q^{th} feedback loop. Note that this TFOM is a set of symbols.)

[ADVISORY]

MCTR Mean Cost To Repair (this measure is based on a fault tree defined as an optimal sequence of tests using the test algorithm)

[MODEL]

MTTR Mean Time To Repair (this measure is based on a fault tree defined as an optimal sequence of tests based on the test algorithm)

[MODEL]

MTTI Mean Time To Isolation (this measure is based on a fault tree defined as an optimal sequence of tests based on the test algorithm)

[MODEL]

MCTI Mean Cost To Isolate (this measure is based on a fault tree defined as an optimal sequence of tests based on the test algorithm)

[MODEL]

MUTI Mean User-defined-function To Isolate (this measure is based on a fault tree defined as an optimal sequence of tests based on the test algorithm)

[GENERAL/MODEL]

TPU_k Test Point Utilization (effectively the probability that the k^{th} test will be used). Note that this TFOM is a vector. It is based on a fault tree defined as an optimal sequence of tests using the test algorithm.

[ADVISORY]

TPC_k Test Point Criticality (a summation of the criticality levels of the subsystems/components that have the k^{th} test in their isolation path). This measure is based on a fault tree defined as an optimal sequence of tests using the test algorithm.

[ADVISORY]

TFOM's IDENTIFIED IN STAMP

IL Isolation Level (the ratio of the number of ambiguity groups to the total number of components). If this were to be weighted based on failure rates and test reliabilities, it would be equivalent to FIR(n) where n is the size of the largest ambiguity group.

[GENERAL]

FMIL Feedback-Modified Isolation Level (this measure is identical to IL with the feedback loops collapsed into single equivalent components)

[ADVISORY]

TL Test Leverage (the ratio of tests to the sum of tests and components)

[GENERAL]

NRTL Non-Redundant Test Leverage (the ratio non-redundant tests to the sum of non-redundant tests and components). Note that in stamp redundant tests are those with identical failure signatures.

[ADVISORY]

FMTL Feedback-Modified Test Leverage (this measure is identical to TL with the feedback loops collapsed into single equivalent components and nested tests removed.)

[ADVISORY]

TU Test Uniqueness (the ratio of non-redundant tests to the total number of tests)

[ADVISORY]

TR Test Redundancy (the complement of TU, i.e., TR = 1-TU)

[ADVISORY]

TFBD Test Feedback Dominance (the ratio of tests involved in feedback loops to the total number of tests)

[ADVISORY]

CFBD Component Feedback Dominance (the ratio of components involved in feedback loops to the total number of components)

[ADVISORY]

NDP Non-Detection Percent (the ratio of the number of components whose failures are not sensed by any tests to the total number of components). Note that if this measure were to be modified to account for failure rates and test reliabilities, it would be the complement of FFD.

[GENERAL]

HFM Hidden Failure Measure (the ratio of the number of components whose failure signatures mask those of other components to the total number of components that could be masked by such failures). This measure reflects the lack of tolerance to a specific class of multiple failures.

[ADVISORY]

IMHFM Input-Modified Hidden Failure Measure (the ratio of the number of components whose failure signatures mask those of other components to the total number of components that could be masked by such failures, discounting the effects of faulty input signals)

[ADVISORY]

PHFM Percent Hidden Failure Measure (the ratio of the sums of cardinalities of the sets of components that could be masked by such failures of other components to its theoretical maximum value). This measure reflects the lack of tolerance to a specific class of multiple failures.

[ADVISORY]

IMPHFM Input-Modified Percent Hidden Failure Measure (the ratio of the sums of cardinalities of the sets of components that could be masked by such failures of other components to its theoretical maximum value, ignoring any masking effects that could be caused by faulty input signals)

[ADVISORY]

FFM False Failure Measure (the ratio of the number of components whose set of multiple failure combinations represent signatures that are identical to those of individual components, to its theoretical maximum value). This measure indicates a lack of tolerance to a class of multiple failures.

[ADVISORY]

IMFFM Input-Modified False Failure Measure (the ratio of the number of components whose set of multiple failure combinations represent signatures that are identical to those of individual components, ignoring the effects of signal inputs, to its theoretical maximum value)

[ADVISORY]

DEP Dependency (the ratio of the summation of higher-order dependencies, on both components and tests, over all tests, to its theoretical maximum value). This measure reflects the overall level of interconnectedness.

[GENERAL]

TIDEP Test interdependency (the ratio of the summation of higher-order test-to-test dependencies to its theoretical maximum value.)

[GENERAL]

TDEP Test dependency (the ratio of the summation of higher-order test-to-component dependencies to its theoretical maximum value.)

[GENERAL]

FAT False Alarm Tolerance (the ratio of the summation of the number of downstream tests that could be used to verify the bad outcome of each test, over all tests, to its theoretical maximum value). This TFOM attempts to measure the overall level of analytical redundancy in the test system that could be used to reduce the occurrence of false alarms.

[ADVISORY]

EXDEP External dependency (the ratio of signal inputs to the sum of components and inputs). This measure is an indication of the degree to which test outcomes may depend on inputs. It also reflects, to some degree, controllability for test.

[ADVISORY]

XM Excess test Measure (the ratio of all redundant and excess tests to the total number of tests). Excess tests are those that may have unique signatures, but represent information that exists with combinations of other tests. This measure is similar to TR.

[ADVISORY]

TFOM's IDENTIFIED IN SCOAP

$CC^0(L)$ Combinational Zero-Controllability of line L (for digital circuits, the minimum number of combinational input line assignments required in order to set the logical value of line L to 0)

[ADVISORY]

$CC^1(L)$ Combinational One-Controllability of line L (for digital circuits, the minimum number of combinational input line assignments required in order to set the logical value of line L to 1)

[ADVISORY]

CO(L) Combinational Observability of line L (for digital circuits, the minimum number of combinational input line assignments required in order to propagate the logical value of line L to a principal output)

[ADVISORY]

$SC^0(L)$ Sequential Zero-Controllability of line L (for digital circuits, the minimum number of sequential input assignments required in order to set the logical value of line L to 0). Sequential input assignments are comprised of combinational input line assignments made over an unspecified number of time frames.

[ADVISORY]

$SC^1(L)$ Sequential One-Controllability of line L (for digital circuits, the minimum number of sequential input assignments required in order to set the logical value of line L to 1). Sequential input assignments are comprised of combinational input line assignments made over an unspecified number of time frames.

[ADVISORY]

SO(L) Sequential Observability of line L (for digital circuits, the minimum number of sequential input assignments required in order to propagate the logical value of line L to a principal output). Sequential input assignments are comprised of combinational input line assignments made over an unspecified number of time frames.

[ADVISORY]

TFOM's IDENTIFIED IN CAMELOT

CTF Controllability Transfer Factor (for digital circuits, the degree of departure from the condition in which it is equally easy to generate both logical 0 and 1 at the principal output of the circuit using the circuit's available inputs)

[GENERAL]

CY(z) Controllability of node z (for digital circuits, a measure of the ability to control the logical value of an arbitrary line, z)

[ADVISORY]

OTF Observability Transfer Factor (for digital circuits, a measure of the ease with which internal line values can be propagated to a principal output of the circuit)

[GENERAL]

OY(I-O) Observability of line I at line O (for digital circuits, a measure of the ease with which the logical value of a given line, I can be propagated to another given line, O)

[ADVISORY]

TY(line) testability of "line" (for digital circuits, the product of line's controllability, CY(line) and observability, OY(line))

[ADVISORY]

TY(C) testability of the digital circuit "c" (for digital circuits, the summation of the testability of all of its lines divided by the number of those lines)

[GENERAL]

TFOM's IDENTIFIED IN STAFAN

C1(L) One-controllability of line L (for digital circuits, the probability that line L will take on a logical value of 1 given an arbitrary test vector applied to its host circuit's input lines)

[ADVISORY]

C0(L) Zero-controllability of line L (for digital circuits, the probability that line L will take on a logical value of 0 given an arbitrary test vector applied to its host circuit's input lines). Note C0(L) = 1 - C1(L).

[ADVISORY]

B0(L) Zero-observability of line L (for digital circuits, the probability that a logical value of 0 on line L can be propagated to a principal circuit output)

[ADVISORY]

B1(L) One-observability of line L (for digital circuits, the probability that a logical value of 1 on line L can be propaga to a principal circuit output)

[ADVISORY]

TFOM's IDENTIFIED IN OPEN LITERATURE

τ-Fault Diagnos-ability [GENERAL]
τ-Fault Diagnosability (a system is said to be τ-fault diagnosable if at least τ faults can be simultaneously diagnosed). This measure takes on values of true or false.

FD/FI [MODEL]
Fault Detection/Isolation (effectively the probability that any fault will be both detected and isolated using a given test methodology/system; FD/FI = FFD * FFI)

FD_c/FI_c [MODEL]
Critical Fault Detection/Isolation (effectively the probability that critical faults will be both detected and isolated using a given test methodology/system). This is typically required to be 100%.

TR_{MAX} [MODEL]
Maximum repair time

$RTOK_{RATE}$ [MODEL]
ReTest OK rate (the time rate at which units sent for shop/depot repair are retest ok without any repair action)

R(D) [MODEL]
Reliability of Diagnosis (the probability that any given diagnostic conclusion is correct)

T_{UD} [MODEL]
Mean first passage time (the expected time required to arrive at the first fault isolation conclusion)

NFRU [MODEL]
Average Number of Field Replaceable Units replaced per detected fault

NSC [MODEL]
Average Number of Service Calls per detected fault

TC [GENERAL]
Test Coverage (the percentage of components in a system that are tested)

C-Test-ability [GENERAL]
C-Testability (for PLA's, the ability to be tested with a constant number of test patterns despite changing array sizes)

I-Test-ability [GENERAL]
I-Testability (for PLA's, the ability to design a test set such that all array cells use the same tests and, if good, yield identical test results)

F_k Information gain per test cost (the expected information gain about component states from the k^{th} test divided by its cost)

[ADVISORY]

I System complexity (the average number of items, e.g., components, modules, or circuit boards, directly connected to any one item)

[GENERAL]

τ_0 Average test time (the average time required to acquire and process information for a single diagnostic step or test)

[MODEL]

DT_N Diagnostic Time (the time required for hypothesis generation and action selection for the N^{th} diagnostic step)

[MODEL]

FRR False Removal Rate (the rate at which good replaceable units are removed as a consequence of maintenance activity)

[MODEL]

J(p,t) Robust redundancy metric (a measure of the tolerance of a system under test to false alarms based upon covariance analyses of the tests)

[ADVISORY]

H(p,t) Robust redundancy metric (a measure of the tolerance of a system under test to false alarms based upon information entropy analyses of the system)

[ADVISORY]

T_ξ Testability measure (a measure of the average rate of information returned on a vector of component values)

[MODEL]

C_ξ Test Complexity (a measure of the time required to acquire the information on a vector of component values to a specified degree of accuracy). This TFOM is computed using T_ξ.)

[MODEL]

2.2.1.2.3 Summary and Commentary on TFOM's Identified.

We have uncovered some 100 reasonably unique testability figures of merit in our literature review and analysis. That number could be inflated in those cases where multiple definitions and/or ambiguous definitions exist. An example of the former is FFD (Fractions of Faults Detected). This particular TFOM has at least two different formulations. An example of the latter, an ambiguously defined TFOM, is FFI (Fraction of Faults Isolated). This figure of merit is an implicit function of a maintenance policy. In a maintenance environment where isolation to 4 or 5 replaceable units is acceptable, we might be able to demonstrate a value of 100% for a given system. In another maintenance environment, that same system might demonstrate an FFI of 70%.

The figures of merit identified were categorized in terms of their usages. Of particular importance to this effort were those that were classified as being useful for analytical modeling. We also categorized the TFOM's using the taxonomic structure formulated in the RADC report by Pliska et al. (see Figure 2.2-3). A cursory comparison of the model-applicable TFOM's against their functional categorizations as given in Figure 2.2-3 indicates that there are sufficient off-the-shelf figures of merit to completely characterize testability. Thus, at this point we can conclude that it is feasible to identify testability figures of merit which can be related to system performance objectives and which completely characterize the testability of that system.

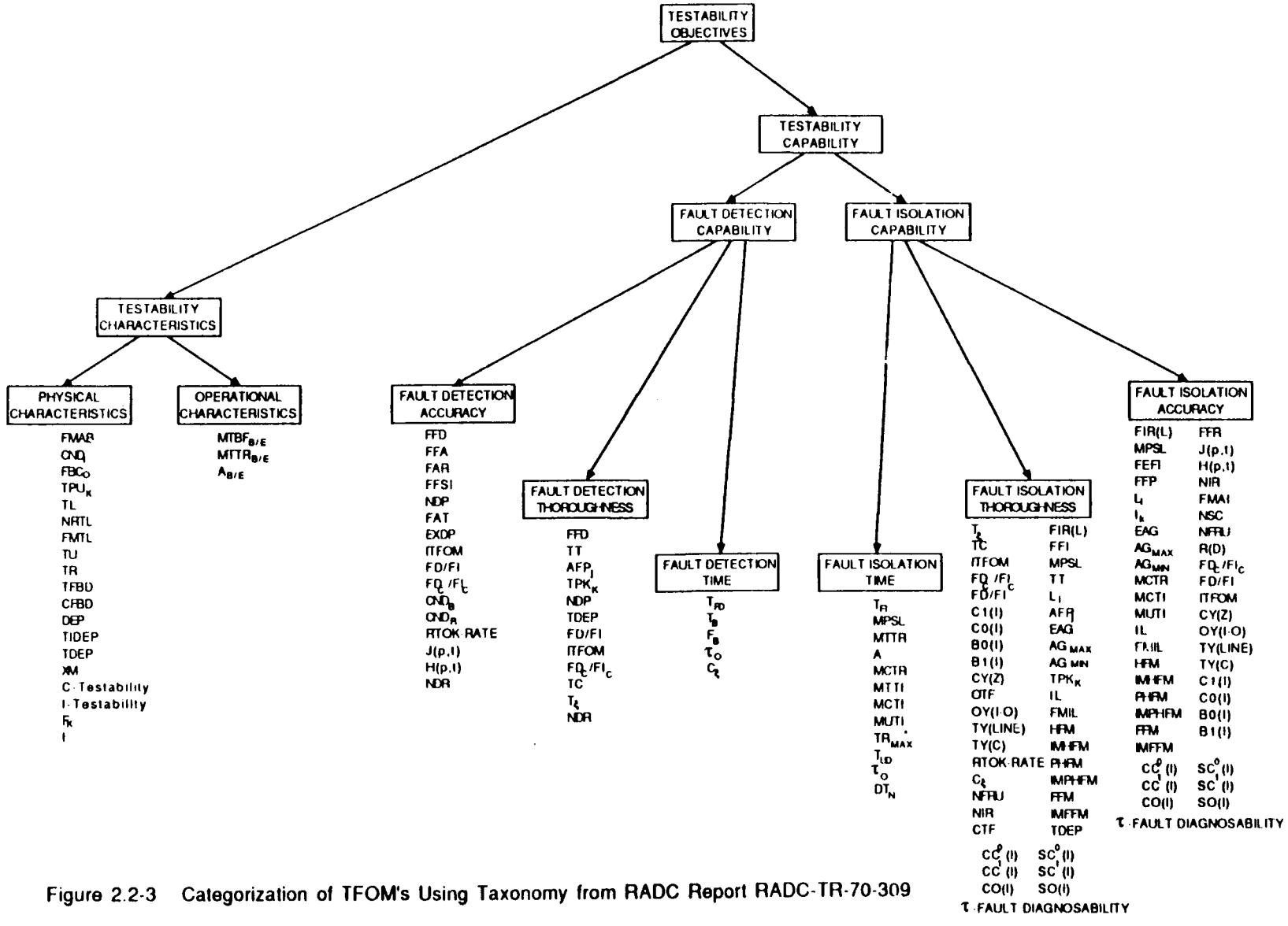


Figure 2.2-3 Categorization of TFOM's Using Taxonomy from RADC Report RADC-TR-70-309

2.2.2 Top-Down Analysis of Testability Requirements.

In the domain of weapon systems there are a number of accepted measures against which system performance may be judged. These measures fall into two categories. The first such category concerns the degree to which a system with no failures can accomplish its intended mission(s). In general, this class of metric will be highly mission specific. They are exemplified by such specifiers as circular error probability, kill ratio, and range. Outside the mission context, those figures have little or no meaning. Since these measures exist outside the realm of failures, there is no relation between them and the incorporation of testability for a given system. The second category of system performance measure is concerned with the effects of failures and the management thereof. This class of metric tends to be insensitive to the context of the weapon system mission(s). Such measures of performance are functions of testability.

The above mentioned performance measures are often weighed against their costs. A design for a given weapon system that exhibits outstanding performance characteristics may be rejected in favor of another design that yields marginally less performance, but costs substantially less. The current trend in assessing such costs accounts for the entire expense of ownership -- both acquisition and operating costs. Thus, the full cost of owning a weapon system is evaluated against its performance. Incorporating testability will have both positive and negative impacts on the costs associated with ownership of a weapon system. Clearly, acquisition costs will increase with the incorporation of testability. However, the costs associated with operation should decline due to savings in the area of maintenance and support.

In this section we will analyze the relationships between measures of performance and testability. Specifically, the parameters,operational readiness and availability will be examined. We will also identify the principal areas of weapon system operating costs that are affected by different testability characteristics. For this analysis, observe that we are ignoring acquisition costs. This is because the testability cost burden represents an objective function to be minimized during allocation and , as such, is treated under that topic. Our primary concern revolves around savings in cost afforded by the incorporation of testability during operation.

2.2.2.1 Operational Readiness.

Operational readiness, P_{or}, defined in Section 1.1.3, is mathematically expressed as:

$$P_{or} = R(T_m) + D * Pr(T_R \leq T_C) [1 - R(T_m)] \quad (2\text{-}20)$$

in which $1 - R(T_m) = Q(T_m)$, the probability that some mission critical failure will occur within the mission duration, T_m. $Q(T_m)$ is a probability that describes two classes of events, real and imaginary failures. Real critical failures are those that would cause a loss of the intended mission. Imaginary failures are artifacts of a diagnostic or test subsystem and are referred to as false alarms. The class of critical false alarms are those that would cause a mission to be aborted even though no failure would actually exist. An example of such an event is the erroneous report of an imminent engine failure by a performance monitor.

If we assume that critical failures and critical false alarms are independent events in the statistical sense, then we can decompose $Q(T_m)$ as follows:

$$Q(T_m) = Q_R(T_m) + Q_I(T_m) - Q_R(T_m)\, Q_I(T_m) \quad (2\text{-}21)$$

where $Q_R(T_m)$ is the probability of an actual mission critical failure within a mission of duration T_m, and $Q_I(T_m)$ is the likelihood of a mission critical false alarm report in that same time frame.

D is the probability, given a mission critical failure occurs, that it will be detected. This term is referred to as the detectability of mission critical faults.

$Pr(T_R \leq T_C)$ is a measure of maintainability. It is defined as the probability that the time required to repair any given failure, T_R, will be less than the allowed checkout time between missions, T_C. This, in turn, is a function of the isolability and repairability of a weapon system. An exponential distribution is typically used.

$$Pr(T_R \leq T_C) = 1 - e^{-T_C/MTTR} \qquad (2\text{-}22)$$

The term MTTR is the mean time to repair. It includes among its constituent times, the expected time to isolation (MTTI).

Relevant Testability Issues

There are several testability-related characteristics that were identified in the above analysis. Specifically, they are:

$Q_I(T_m)$ - The probability of erroneously reporting a mission critical failure during the course of a mission -- a mission critical false alarm

D - The probability of detecting a mission critical failure given that one occurs

MTTI - The expected time to isolation of a fault

2.2.2.2 Availability.

The availability of a weapon system is defined as the probability that the system will be operational at some time, say τ_1, given that it was operational at a prior time τ_0. Mathematically, availability is expressed as:

$$A = \frac{MTBF}{MTBF + MTTR} \qquad (2\text{-}23)$$

where MTBF is defined as the mean time between failures and MTTR, as was mentioned above, is defined as the mean time to repair. Further, by definition we have the following:

$$MTBF = 1/\Lambda \qquad (2\text{-}24)$$

where Λ is the system failure rate (assuming no fault tolerance in the design). In the previous analysis of operational readiness, we decomposed

failures into those that are real and those that are imaginary, (artifacts of a diagnostic subsystem). Similarly, we may decompose Λ into its real and imaginary components.

$$\Lambda = \lambda + FAR \qquad (2\text{-}25)$$

λ is the rate of occurrence of real failures and FAR is the rate of occurrence of false alarms.

In this scenario, the effective mean time to repair, MTTR', is comprised of two components, the average time required to repair faults and the average time required to clear false alarms, MTTR and MTFA, respectively.

$$MTTR' = (\lambda / \Lambda)\, MTTR + (FAR / \Lambda)\, MTFA \qquad (2\text{-}26)$$

As we observed in our discussion on operational readiness, MTTR is comprised of a series of task times that include MTTI.

The average time to clear false alarms, MTFA, is difficult to estimate. In some cases, false alarms will be immediately recognized for what they are and dismissed. In this event no time is considered to have been spent to clear the alarm. In some cases, they will result in a brief fault detection checkout or retest to verify the permanence of the fault. In this case, we can estimate the time spent as the average time required for detection, T_D.

Relevant Testability Issues

There are several testability-related characteristics that were identified in the analysis of availability. They are:

FAR - The rate at which false alarms occur, (observe that FAR / Λ is the conditional probability of the occurrence of a false alarm, given that a fault has been reported)

T_D - The average time required to detect a failure given that one has occurred

MTTI - The expected time to isolation of a fault

2.2.2.3 Operating Costs.

A simple model for the entire life cycle cost associated with weapon system ownership is the sum of the non-recurring (acquisition related) and recurring (operating) costs, NRC and RC, respectively.

$$LCC = NRC + RC \tag{2-27}$$

The components that make up the non-recurring costs are:

- C_{RD} - Research and Development Costs
- C_{RM} - Reliability and Maintainability Improvement Costs
- C_Q - Qualification Approval Costs
- C_{LCM} - Life Cycle Management Costs
- C_A - Acquisition Costs
- C_I - Installation Costs
- C_{TE} - Test Equipment Acquisition Costs
- C_T - Training Costs

Clearly, a number of those non-recurring cost elements will be functions of testability. In general the more and better the testability, the higher the acquisition-related costs will be.

In our prior analyses we sought to identify relationships between testability concerns and measures of weapon system performance. As a consequence of this perspective, we are also concerned with the sensitivity of operating costs to testability issues.

These costs are comprised of the following elements:

- C_O - Normal Operating Costs
- C_M - Manpower Costs
- C_S - Support Costs
- C_{MT} - Maintenance Costs
- C_{IN} - Inventory Costs

Of these terms, the costs directly impacted by the incorporation of testability are, maintenance/manpower costs, and inventory costs. The

maintenance/manpower costs are driven primarily by the staffing expenses necessary to maintain the weapon system. On the other hand, inventory costs are primarily driven by the expense of supplying spare parts. In the sections that follow we will briefly analyze these cost elements. Our analysis will focus on the restricted case of a single weapon system at a single maintenance level.

2.2.2.3.1 Maintenance Costs.

The principal contributors to the maintenance costs within a given maintenance level are:

Labor costs for fault detection
Labor costs associated with false alarms
Labor costs for fault isolation
Labor costs for fault rectification

Each of these can, in turn be evaluated in terms of testability.

Labor Costs for Fault Detection

The labor required for fault detection will be a function of the average time required for the detection process within the scope of some pre-defined maintenance scenario. In addition, it will be sensitive to the percentage of faults detected within that scenario.
Note that the undetected faults (i.e., 1-FFD) have no associated labor costs. This is because those faults are detected outside the specified testing process, often during a mission, by non maintenance staff.

Labor Costs Associated with Occurrence of False Alarms

The expenses associated with the occurrence of false alarms, must include the labor costs of acquiring the false alarm reports, as well as those for the clearance of the alarms. In addition, these costs are sensitive to the rate of occurrence of false alarms, FAR.

Labor Costs for Fault Isolation

The costs associated with fault isolation are sensitive to the mean time to isolation, MTTI, and to the hourly labor cost involved with the isolation process. This also clearly depends on the maintenance strategy, e.g. sequential or block replacement.

Labor Costs for Fault Rectification

Fault rectification consists of the replacement of faulty LRUs and the subsequent checkout of the weapon system. Factors relating to maintainability as well as testability play a role in these costs.

2.2.2.3.2 Inventory Costs

It appears that a monotonic relationship exists between the average number of units replaced for every failed unit. In an idealized logistics environment we can assume that the spares are adequately replenished at regular time intervals. For this scenario, we can estimate that the costs associated with this inventory on a per system basis are sensitive to the following factors:

- the cost of maintaining an item in inventory
- the cost of placing an item in inventory (per unit time)
- the number of spares (which depend on failure rate, and on the expected number of removals per failed unit).

As other factors such as the effect of repair pipelines are introduced, inventory costs will increase.

2.2.2.3.3 Testability Concerns.

From our very cursory evaluation of the various operating costs associated with weapon system ownership, we can identify several specific testability features that impact those costs. They are percentage of faults detected, average detection time, rate of occurrence of false alarms, average isolation time, average number of units replaced per failed unit.

2.2.2.4 Summary of Relevant Testability Issues.

In the course of the above analyses of system performance requirements and system operating costs, we have been able to establish analytical relationships between testability characteristics and the system performance/costs. It is encouraging to note that several of the testability characteristics are related to most of the performance specifiers and operating costs. The testability characteristics resulting from those analyses are:

Fault Detection Probabilities
False Alarm Probabilities and Rates
Fault Detection Times
Fault Isolation Times
Fault Isolation Resolution per Failed Unit

These characteristics represent a bridge with which to connect our eventual set of TFOM's to the various performance measures and costs. Thus, as the set of TFOM's is selected, we must ensure that all of the above characteristics can be derived.

2.2.3 Selection and Refinement of TFOM's.

The set of TFOM's that was identified and reported in Section 2.2.1 must now be reduced to some minimal subset. The goal now is to select a subset and modify its members as necessary for subsequent application in the TAM. The criteria for that selection, based on the objectives of this effort, are various. In the following paragraphs those criteria are presented in the form of filters for reducing the set of TFOM's reported in Figure 2.2-3. Along with the descriptions of the filters, the results of the filtering processes are given.

FILTER 1: COVERAGE	*The set of TFOM's selected must represent a maximum amount of coverage of the testability characteristics.*

The testability capabilities identified in the RADC Report by Pliska et al. are fault detection and isolation. Each of them are, in turn, broken into the attributes of thoroughness, accuracy, and time. Thus we begin the filtering process by eliminating other concerns, namely, those associated with system acquisition This reduction is illustrated in Figure 2.2-4. This filter now becomes a constraint. We have to assure ourselves that the set of TFOM's has members in all of the remaining categories.

FILTER 2: RELATABILITY TO GLOBAL PERFORMANCE MEASURES	*The individual TFOM's selected must be traceable to such global measures of performance as availability, mission dependability, and life-cycle costs. More esoteric measures can seldom be related to real-life performance measures.*

The testability figures of merit identified in Section 2.2.1.2.2 that were categorized as being applicable to modeling pass this filter. This second reduction is illustrated in Figure 2.2-5. As was the case with the first, this filter now becomes a constraint. We must assure ourselves that the final set of TFOM's can be used to generate all of the attributes identified in our top down analysis (Section 2.2.2.4).

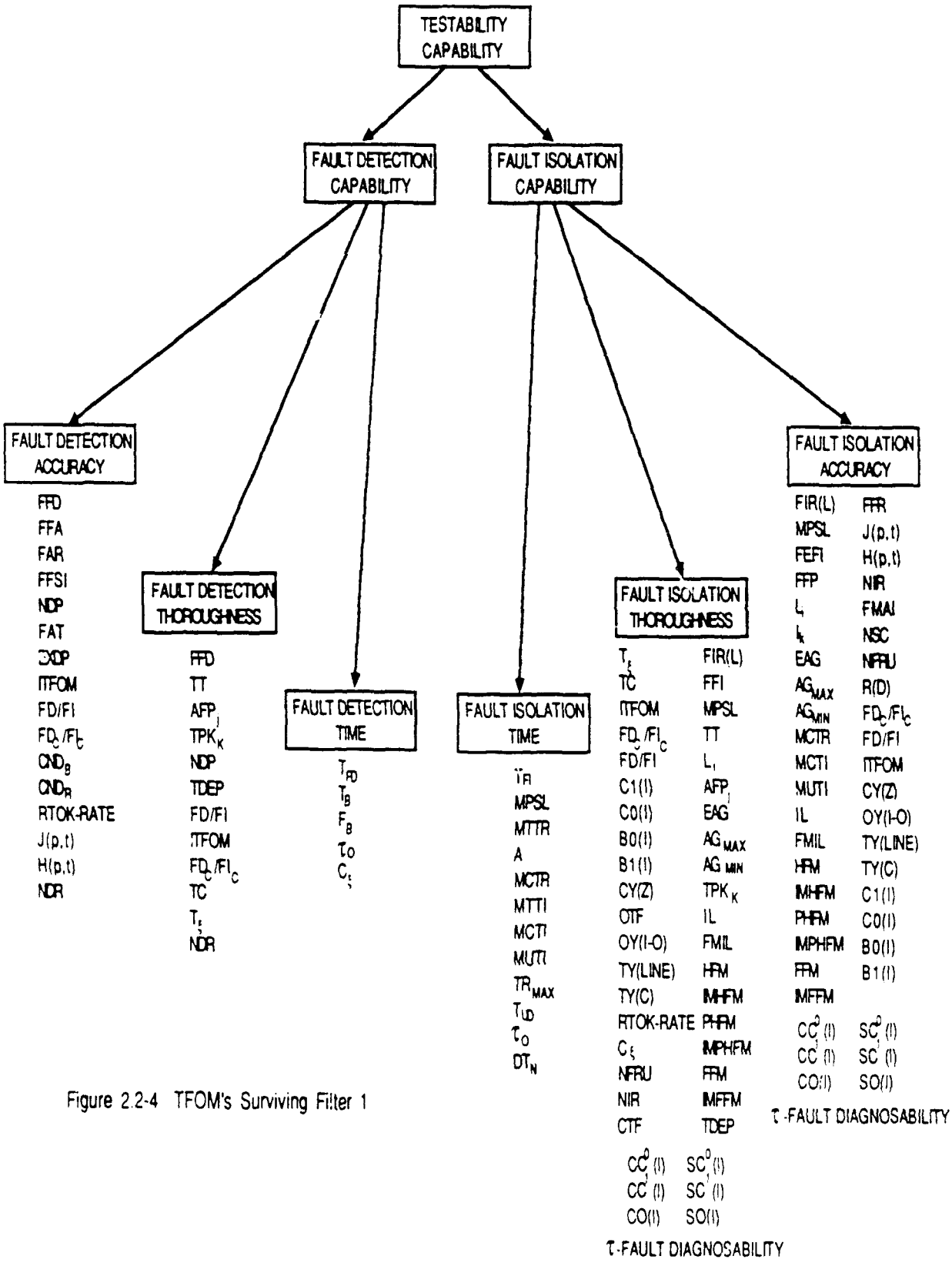


Figure 2.2-4 TFOM's Surviving Filter 1

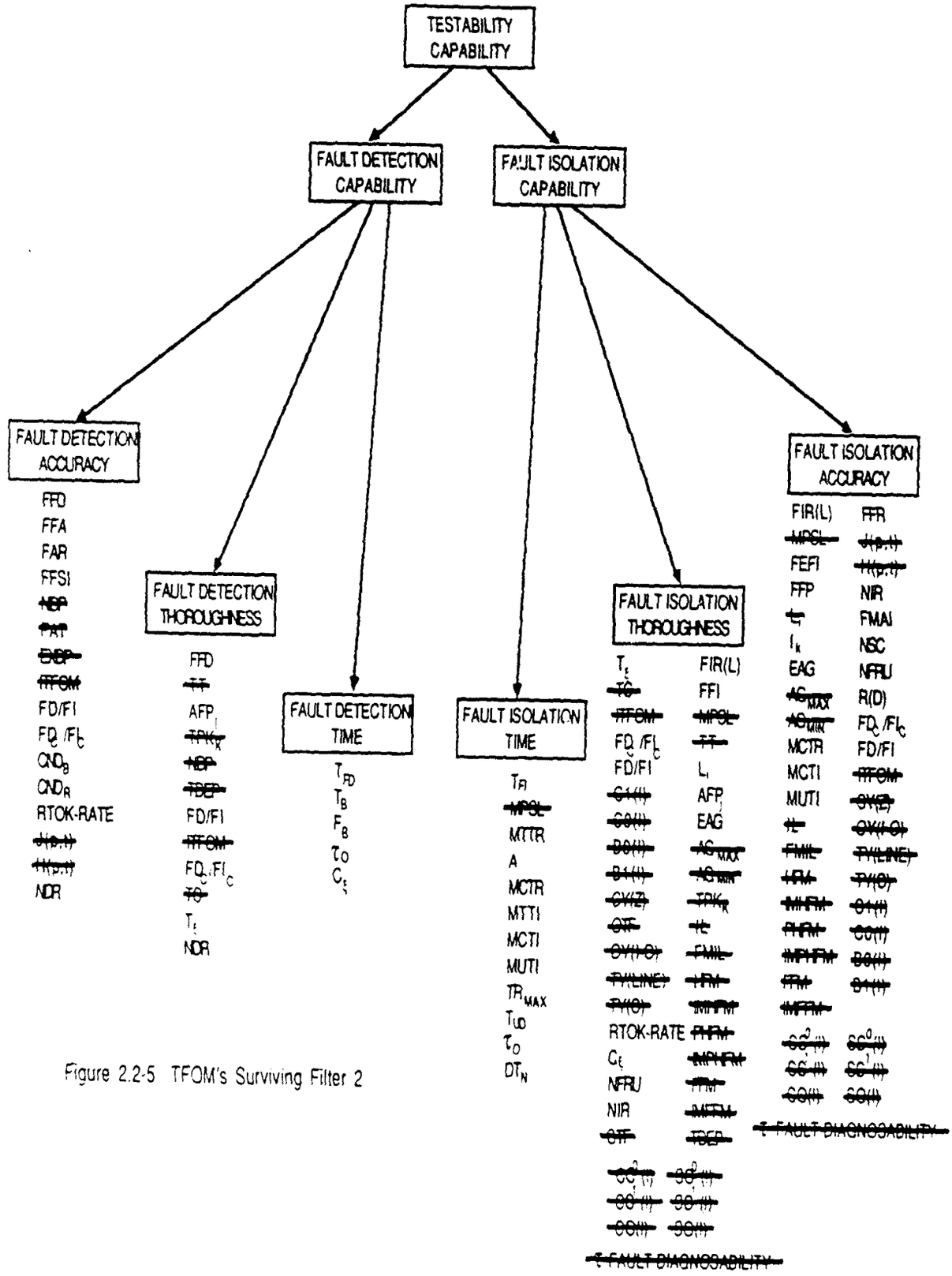


Figure 2.2-5 TFOM's Surviving Filter 2

FILTER 3: AMBIGUITY	*The TFOM's selected should be unambiguous in their definitions. No room for interpretation should exist. The figures should not vary as functions of any operational issues such as defined maintenance strategies. As an example, the definition of FFI varies as a function of "maintenance concept."*
FILTER 4: ORTHOGONALITY	*The set of measures selected must represent, as nearly as possible, an orthogonal set. That is to say each TFOM should be a measure of an independent characteristic.*
FILTER 5: LEVEL OF SPECIFICATION	*The set of TFOM's selected should represent the absolute minimum level of specification necessary to completely specify testability performance. Too much specification constrains the design and thus minimizes opportunities for innovation.*

The testability figures of merit remaining after the second filter were further reduced by filters 3, 4, and 5. The results of this reduction are illustrated in Figure 2.2-6.

FILTER 6: ASSESSABILITY	*The TFOM's selected must be calculable from design data -- both at PDR and CDR stages of design. To the extent possible, they should be field verifiable.*
FILTER 7: TRADITION	*The TFOM's should, when possible, be chosen from among those accepted in the military design community. Those that are new should be stated in terms that relate them to the more traditional TFOM's*

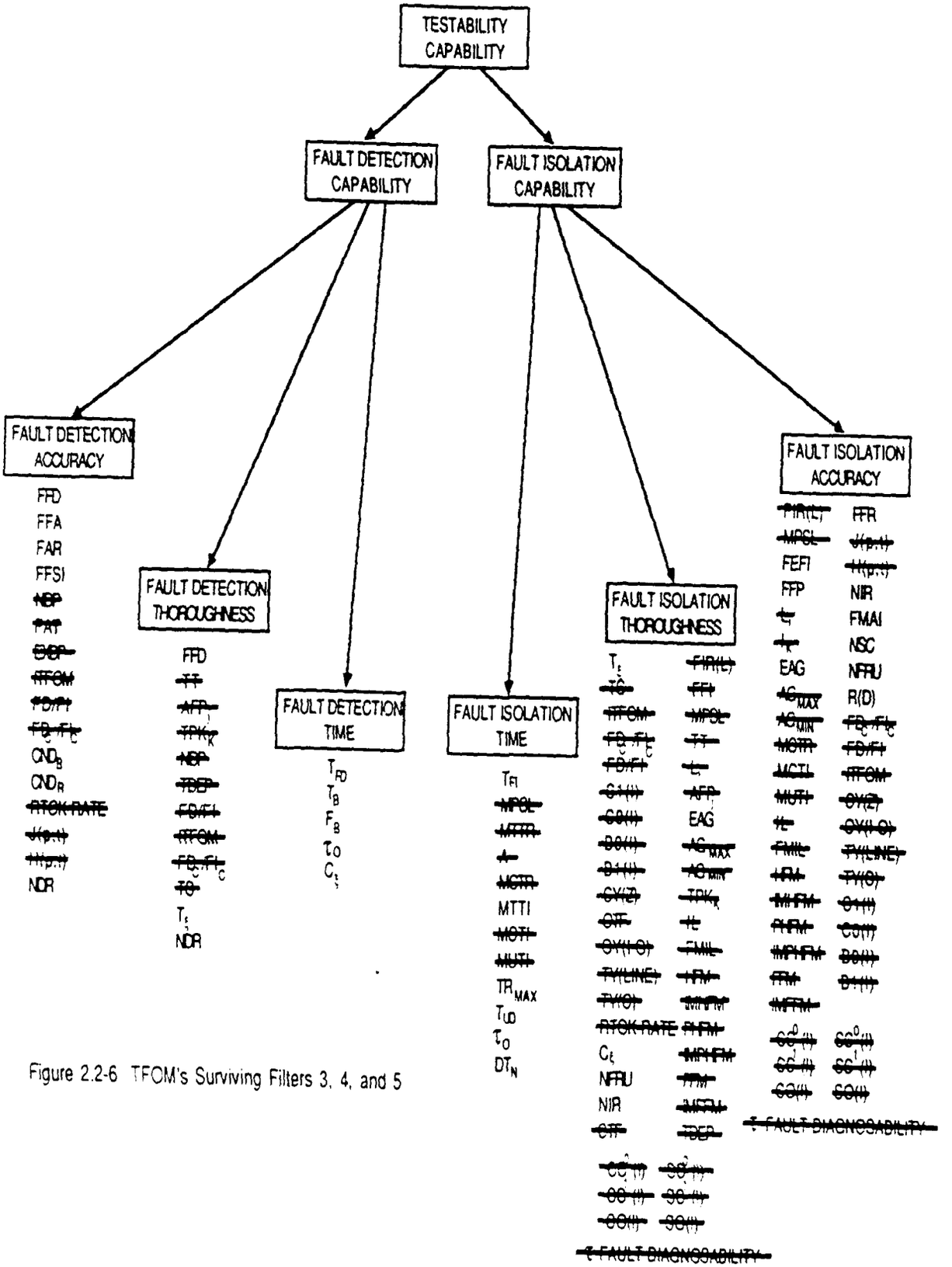


Figure 2.2-6 TFOM's Surviving Filters 3, 4, and 5

The testability figures of merit remaining after the third, fourth, and fifth filters were further reduced by filters 6 and 7. At the conclusion of this level of filtering, several instances of multiple TFOM's in the individual categories remained. At that point all seven filters were re-applied to this small set of TFOM's yielding a final set of six.

The results of this reduction are illustrated in Figure 2.2-7. With some minor refinements, we ended up with a set of six figures of merit that are, in essence, "off-the-shelf" and estimable from design data. Further, the members of the set are all potentially field verifiable. The six TFOM's are:

FD - Fraction of Faults Detected
FA - Fraction of False Alarms
T_D - Mean Time for Detection
FI_P - Fractional Isolability
FP - Fraction of False Pulls
T_I - Mean Time to Isolation

Each of the above TFOM's is defined and analyzed in the following sections.

2.2.3.1 FD - Fraction of Faults Detected.

Within the scope of some test environment (e.g., BIT, ETE, BIT and ETE, etc.), FD is defined as the fraction of (all) faults detected. Its range is between 0 and 1 (or 0% to 100%). This figure of merit measures both the breadth of fault coverage and the goodness of the tests. For example, BIST (Built-In-Self-Test) in a RAM device may only detect faults 90% of the time. Thus, a failure of that device is only partially detected. If we hypothesize a circuit board comprised entirely of those RAM's, the FD for the board will be 90%, even though we test 100% of the devices. Only if all tests were perfect, would our board-level FD be 100%.

The definition for FD, as an in-the-field measure, is the fraction of (all) faults detected by the test environment under consideration.

$$FD = Q_{DF}/Q_F \qquad (2\text{-}28)$$

where Q_{DF} is the quantity of faults detected and Q_F is the quantity of all faults, exclusive of false alarms.

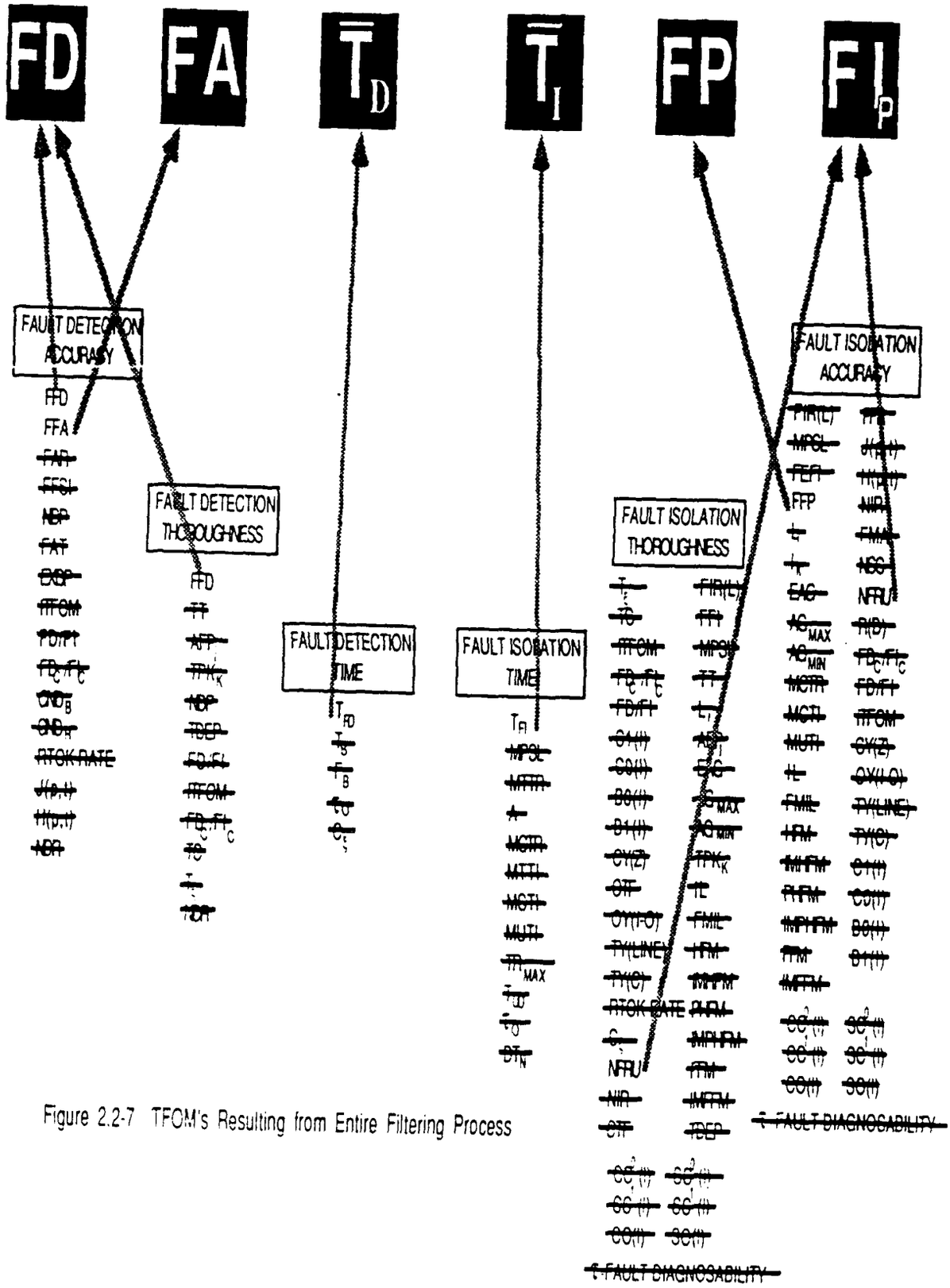


Figure 2.2-7 TFOM's Resulting from Entire Filtering Process

The more general definition of FD is the probability that any given fault will be detected.

$$FD = \sum_{i=1}^{\text{number of modes}} \Pr\{\text{FAILURE MODE i OCCURS}\}\ \Pr\{\text{FAILURE MODE i DETECTED} \mid \text{FAILURE MODE i OCCURS}\} \quad (2\text{-}29)$$

where number of modes is the number of potential failure modes in the system under evaluation.

The conditional probability Pr{FAILURE MODE i DETECTED | FAILURE MODE i OCCURS} can, in turn, be estimated.

$$\Pr\{\text{FAILURE MODE i DETECTED} \mid \text{FAILURE MODE i OCCURS}\} = \Pr\left\{ \bigcup_{test=1}^{\text{number of tests}} (\text{FAILURE MODE i DETECTED BY test} \mid \text{FAILURE MODE i OCCURS}) \right\} \quad (2\text{-}30)$$

where the symbol **U** represents the union of events. In this case, the individual events are the conditional probabilities that the individual tests in the test environment will detect the occurrence of a specific failure.

2.2.3.2 FA - Fraction of False Alarms.

FA is defined as the fraction fault detection reports that are false alarms (in a given test environment). The range of FA is between 0 and 1. This class of events is entirely an artifact of the tests that comprise the test environment. False alarms may be any failure reports due to faulty BIT, ETE, or human observations, out-of-tolerance conditions, or transient conditions. Intermittent and transient faults are not classified as false alarms. FA is observationally defined by the following quotient:

$$FA = Q_{FA}/Q_{FR} \quad (2\text{-}31)$$

where Q_{FA} is the quantity of false alarms, and Q_{FR} is the quantity of failure reports that includes both real and imaginary failures (i.e., false alarms).

This equation can be rewritten in terms of rates as

$$FA = \frac{FAR}{FAR + \lambda * \bar{FD}} \qquad (2\text{-}32)$$

where FAR is the rate of occurrence of false alarms and λ is the rate of occurrence of real failures. Observe that the above expression is easily inverted to yield FAR.

$$FAR = \frac{FA}{1 - FA} \lambda * FD \qquad (2\text{-}33)$$

The more general definition of FA is the probability that any given fault indication is erroneous. There are two independent classes of false alarms, those caused by unstable tests, and those caused by external errors. Thus, we have

$$FA = FA_T + FA_S - (FA_T\, FA_S) \qquad (2\text{-}34)$$

where FA_T is the probability that any test used in the fault detection system under evaluation will be unstable and issue a false positive response (i.e., test fails when there is no failure) and FA_S is the probability that a test in that system will test positive due to an external error, e.g., an erroneous signal at a principal input. The former term is derived from evaluating the quality of the tests in the system, while the latter is based on a topological analysis of the system under evaluation.

The analysis required to estimate FA_S can be performed by many of the testability tools, including IDSS/WSTA. We can describe this class of false alarms as the intersection of three independent events. They are 1) an externally caused error propagates into the system under consideration via a principal input, 2) the presence of the error is detected by one or more tests, and 3) the error is ambiguous with some internal system failure. Thus, the probability of a systemic false alarm, FA_S, is

$$FA_S = Pr\left\{ \bigcup_{j=1}^{\text{number of inputs}} \left[\begin{array}{l} \text{(INPUT j ERRONEOUS) } \textit{AND} \\ \text{(INPUT j ERROR DETECTED) } \textit{AND} \\ \text{(INPUT j ERROR AMBIGUOUS WITH INTERNAL FAILURE MODES)} \end{array} \right] \right\} \quad (2\text{-}35)$$

FA_T is simply the likelihood that any test in the environment under consideration fails given that no fault exists.

$$FA_T = Pr\left\{ \bigcup_{test=1}^{\text{number of tests}} \text{(test i FAILS | NO FAULT EXISTS)} \right\} \quad (2\text{-}36)$$

The value of FA derived from this analysis represents a lower bound on cannot duplicate (CND) events, which are easily measured in the field.

2.2.3.3 T_D - Mean Detection Time.

T_D is defined as the expected time required for the fault detection process, which is similar to Mean Fault Detection Time T_{FD}, defined in section 2. Mathematically, T_D is formulated as follows:

$$T_D = \sum_{i=1}^{Q_{DF}} Pr\{\text{FAILURE MODE i OCCURS}\}\, t_i \quad (2\text{-}37)$$

where t_i is the time required to detect the i^{th} detectable fault, and Q_{DF} is the quantity of detectable faults. Observe that this measure accounts for tests in both process monitoring, and scheduled or on-demand test environments.

In the case of the process monitoring environment, the effective test times are computed based on the frequencies of their executions. The detection times, t_i, are then the expected fault latencies. That is to say, in the performance monitoring environment, the detection time for the i^{th}

fault is the expected time between its occurrence and its observation by a test.
In the case of the scheduled or on-demand test environment (e.g., ground-based check out), the actual test times are employed in the calculation of the detection times. Thus, the expected time for detection is the average time required to sense the existence of a fault, given that one exists, from the point at which the test process is initiated.

2.2.3.4 FI_P - Fractional Isolability

Within the scope of a given test and maintenance environment, FI_P is defined as the fraction of failed units replaced as a consequence of isolation procedures. This FOM is the inverse of one used in industry, NFRU[1]. The theoretical range of FI_P is from 0 to 1; however, in practice, the range is limited to $(\text{number of units})^{-1}$ to 1, where number of units is the number of replaceable units that comprise the system under evaluation.

FI_P is particularly interesting in that it recognizes partial isolations. A partial isolation may be defined as any isolation to an ambiguity group of size greater than 1. This concept is intuitively pleasing if one considers the isolation process as the successive pruning of an ambiguity group that, in the beginning, is of the size number of units. Ideally, the pruning continues until the size of the remaining ambiguity group is one. Any stoppage of the process prior to reaching that objective results in a partial isolation.

The field-observational definition of FI_P is

$$FI_P = Q_{FU} / Q_{RU} \qquad (2\text{-}38)$$

where Q_{FU} is the number of failed units discovered in a specified period of time and Q_{RU} is the number of units replaced in that same time interval. Under the current maintenance structure, this value can be estimated for the organizational level by monitoring the events at the next lower maintenance level. This equation can be rewritten in terms of rates as

1 NFRU is defined as the average Number of Field Replaceable Units replaced per detected fault. It was reported by Bossen et al. in "Model for Transient and Permanent Error Detection," *IBM Journal of Research and Development*, Vol 26, No. 1, January 1982.

$$FI_P = \frac{\dot{Q}_{RU} - \dot{RTOK}}{\dot{Q}_{RU}} \qquad (2\text{-}39)$$

where $\dot{Q}_{RU}$ and $\dot{RTOK}$ are the time rates at which units are replaced at a specified level of maintenance and sent to a lower maintenance level for repair, and which the units that have been sent for repair are returned to inventory without having any faults discovered, respectively.

The design data based definition for FI_P is

$$FI_P = E\{F\} / E\{AG\} \qquad (2\text{-}40)$$

where E{F} is the expected number of failures (usually assumed to be 1), and E{AG} is the expected ambiguity group size. The factor E{F} can be adjusted upward to account for potential multiple failures, or downward to account for expected false alarms. E{AG} is estimated as follows

$$E\{AG\} = \sum_{ag=1}^{\text{number of groups}} Pr\{\text{FAILURE IN AMBIGUITY GROUP ag}\}\, S_{ag} \qquad (2\text{-}41)$$

where number of groups is the number of ambiguity groups of replaceable units created by the test environment under evaluation, FAILURE IN AMBIGUITY GROUP ag is the event that a replaceable unit in the ag^{TH} ambiguity group fails, and S_{ag} is the number of replaceable units that comprise the ag^{TH} ambiguity group. In using the above expression we assume that only one ambiguity group in our system has experienced a failure. This assumption is not unreasonable. It can be argued that all failures, both single and multiple, can be partitioned into a set of mutually exclusive events. The ambiguity groups may then be comprised of combinations of, mutually exclusive , single and multiple failure events. The likelihood of a failure in ambiguity group ag is

$$\Pr\{\text{FAIULRE IN AMBIGUITY GROUP ag}\} = \sum_{i=1}^{\text{modes in ag}} \Pr\{\text{FAILURE MODE i OCCURS}\} \qquad (2\text{-}42)$$

where modes in ag is the number of failure modes that constitute ambiguity group ag.

2.2.3.5 FP - Fraction of False Pulls

FP is defined as the fraction of false pulls. The range of FP is between 0 and 1. More precisely, FP is the probability that any given unit removed is not faulty. False pulls may also be thought of as isolation of imaginary faults.

In terms of fielded data, FP is defined as follows

$$FP = (Q_{RU} - Q_{FU})/Q_{RU} \qquad (2\text{-}43)$$

where Q_{RU} is the number of units removed and Q_{FU} is the number of failed units. Note that the above expression can be reformulated as

$$FP = 1 - (Q_{FU}/Q_{RU}) = 1 - FI_P \qquad (2\text{-}44)$$

Thus, TFOM's FI_P and FP are complements of one another. Because of this relationship, one of them is redundant.

We compute FP from design data using the previous expression for FI_P

$$FP = (E\{AG\} - E\{F\}) / E\{AG\} \qquad (2\text{-}45)$$

Under the single failure assumption and assuming there will be no test errors, the expression reduces to

$$FP = (E\{AG\} - 1) / E\{AG\} \qquad (2\text{-}46)$$

The assessment of E{AG} was described in Section 2.2.3.4 and is therefore

not repeated here.

As is the case with false alarms, there are two independent causes of false pulls, namely, lack of resolution due to insufficient test coverage and inaccurate tests. We can decompose FP on that basis.

$$FP = FP_T + FP_R - (FP_T\, FP_R) \tag{2-47}$$

where FP_T is the probability that, due to an erroneous test outcome, a good part is accused of being faulty and is thus replaced. FP_R is the probability that, due to lack of test resolution, a good part will be replaced along with the faulty part. The former term is derived from evaluating the quality of the tests in the system, while the latter is based on a topological analysis of the system under evaluation. That type of analysis is performed by many of the testability tools, including IDSS/WSTA.

2.2.3.6 T_I - Mean Isolation Time

T_I is defined as the average time required to isolate a fault in the test environment under evaluation. It is also referred to as MTTI. T_I is evaluated by analyzing the tests in the system under evaluation. It is verifiable by fault insertion and exhaustive time studies. Its formulation follows.

$$T_I = \sum_{ag\,=\,1}^{\text{number of groups}} \Pr\{\text{FAILURE IN AMBIGUITY GROUP ag}\}\, t_{lag} \tag{2-48}$$

where number of groups is the number of ambiguity groups of replaceable units created by the test environment under evaluation, FAILURE IN AMBIGUITY GROUP ag is the event that a replaceable unit in the ag^{TH} ambiguity group fails, and t_{lag} is the time required to isolate to the ag^{TH} ambiguity group. t_{lag}, in turn, is determined by summing the times of those tests that comprise the isolation strategy path to arrive at the conclusion that a failure exists in the ag^{TH} ambiguity group.

$$t_{lag} = \sum_{test\,=\,1}^{\text{tests in path to ag}} t_{test} \tag{2-49}$$

where t_{test} is the time required to run test and tests in path to ag is the number of tests in the isolation path to ambiguity group ag. Note that the value of T_I is highly dependent on the size of the ambiguity groups allowed.

2.2.3.7 Summary and Commentary on TFOM's Selected.
Approximately 100 testability figures of merit were analyzed and systematically filtered down to a set of six. Of those six, two are redundant with each other, specifically, FI_P and FP. The TFOM's describe the testability characteristics that are required to support performance and LCC models (Section 2.2.2.4). The set also covers the testability performance categories identified by Pliska et al. (Figure 2.2-7). In summary we have met all of our objectives for identifying a minimal yet complete set of testability descriptors that are suited for allocation by the ATDT TAM.

Of the six TFOM's that were selected, three describe detection and three describe isolation. The two sets are each measures of three inherently orthogonal concepts,

real detections/isolations
imaginary detections/isolations
process time for detection/isolation

The relationships between the sets are graphically demonstrated in Figure 2.2-8. The symmetry between these two sets of TFOM's along with their alignment with inherently orthogonal axes makes them intuitively pleasing. Because of the dual relationships between detection and isolation, (i.e., detection at one level of indenture constitutes isolation at another level of indenture) the TFOM's are insensitive to the level of indenture. That is to say, they can be used at all levels of system indenture.

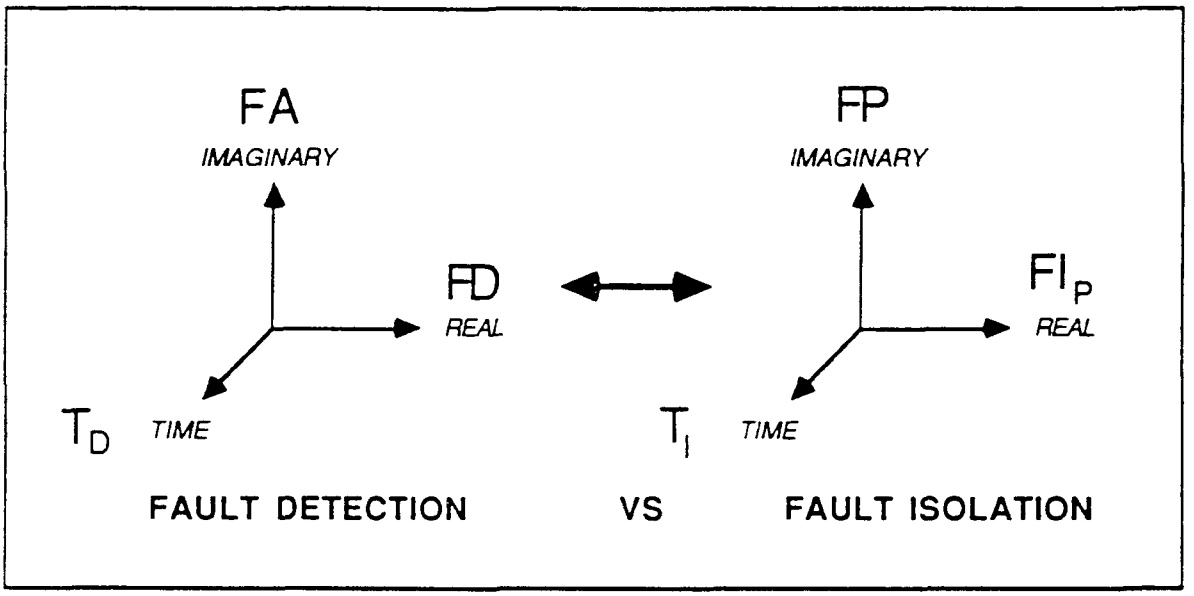


Figure 2.2-8 Graphical description of relationships between TFOM's as descriptions of detection and isolation

2.2.4 TFOM Translation Analysis.

In the previous sections we described the systematic selection of a set of six testability figures of merit. The ATDT TAM will have as its objective the allocation of those TFOM's to descending levels of system indenture. In order to accomplish such an allocation, a procedure is necessary to verify that allocation objectives have been met. (See Figure 2.2-9.) For example, if we use our TAM to allocate TFOM's from the system to the LRU, then we must later be able to verify that our objectives have been met.

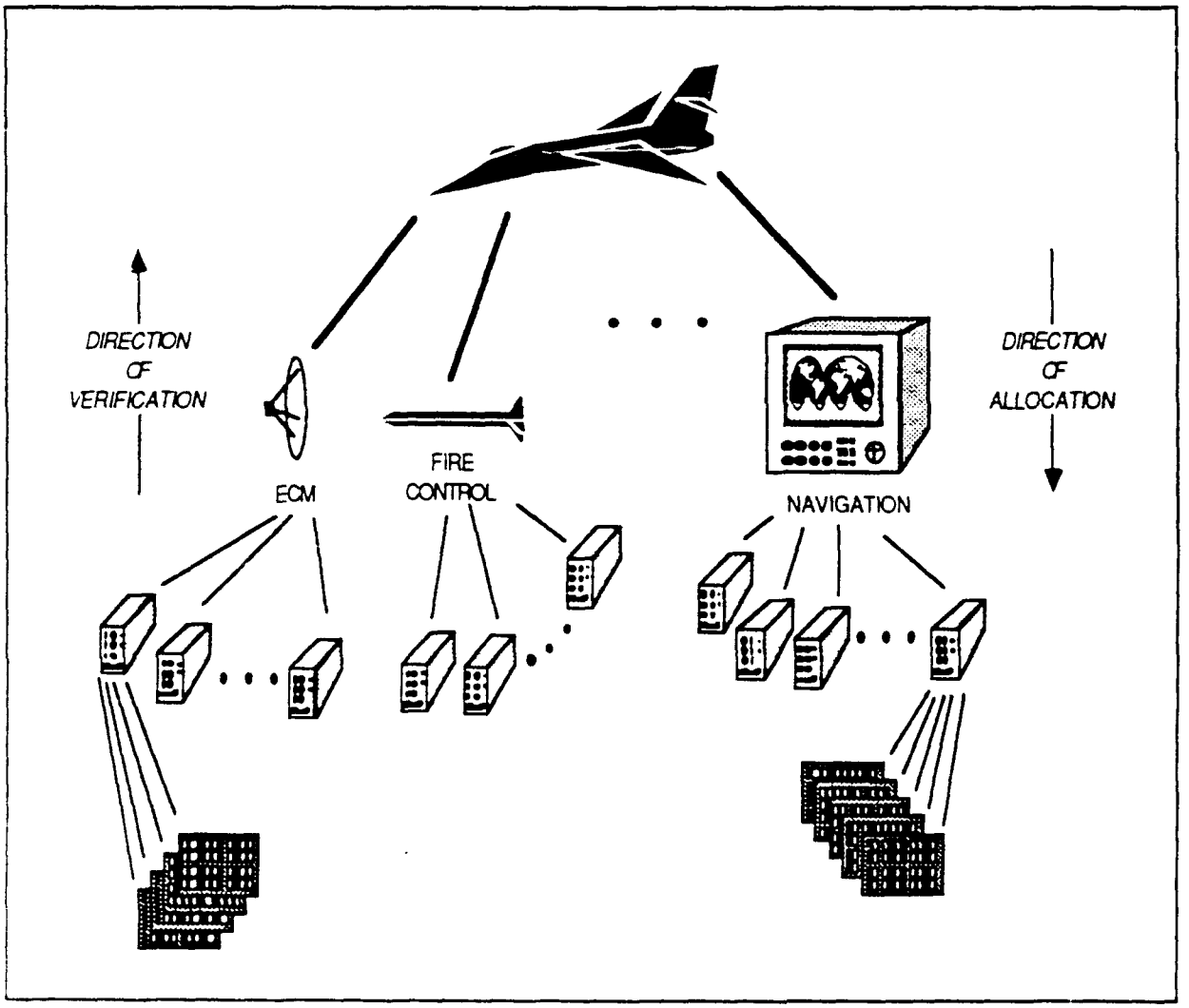


Figure 2.2-9 ATDT top-down testability allocation methodology (TAM) versus bottom-up verification methodology of combining TFOM's (from lower to higher levels of indenture)

Such a verification would involve two steps. The first step would be to compute the TFOM's for each LRU from design or fielded data. This topic was discussed in Section 2.2.3. In the second step the figures of merit for all LRU's would then be combined to produce the TFOM's for the system. It is this combination of TFOM's that is the objective of this level of TFOM translation analysis. This type of analysis is performed for each of the six TFOM's in the following sections.

For the purposes of this discussion we define two arbitrary levels of system indenture. The entities that comprise the lower of the two levels will be called elements. The entities that comprise the higher of the two levels of indenture will be called assemblies. We will use the subscript "i" to indicate the i^{th} element in the elemental level of indenture. The subscript "A" will be used to indicate a value at the assembly level of indenture. Finally, we will use the subscript "+" to indicate a combination of values at the assembly level of indenture. Elements can represent components, SRU's, subsystems. etc. Similarly, assemblies represent circuit boards, LRU's or systems, respectively. For each of the six TFOM's methods are derived for the simple combination of elemental TFOM values to form assembly level values (for example combining values FD_i to form FD_+). The derivations are then extended to accomodate the addition of test capabilities at the assembly level. For example, we combine FD_i from the elemental level and FD_A from the assembly level to form the aggregate value FD'_+.

2.2.4.1 Combining FD

The rate at which elemental detections are reported FDR_i is a function of the rate at which failures in a given element occur, λ_i , and the probability that any given fault in element i will be detected, FD_i. We estimate FDR_i as follows

$$FDR_i = FD_i \lambda_i \qquad (2\text{-}50)$$

The rate at which fault detections in all elements at a given level occur will be the summation of the individual detection rates

$$FDR_+ = \sum_{i=1}^{\text{no. of elements}} FDR_i = \sum_{i=1}^{\text{no. of elements}} FD_i \lambda_i \qquad (2\text{-}51)$$

where no. of elements is the number of elements that comprise a given assembly. Using the same reasoning as we did for FDR_i, we can state that

$$FDR_+ = FD_+ \lambda_+ \qquad (2\text{-}52)$$

Further, the failure rate for the assembly is simply the sum of those of its constituent elements

$$\lambda_+ = \sum_{i=1}^{\text{no. of elements}} \lambda_i \qquad (2\text{-}53)$$

Combining the last three expressions, we can compute FD_+ as:

$$FD_+ = \sum_{i=1}^{\text{no. of elements}} (\lambda_i / \lambda_+) FD_i \qquad (2\text{-}54)$$

This technique for combining the elemental values of FD into an assembly level TFOM is known as failure rate transformation. This form of transformation works between any two levels of indenture.

We now complicate the picture by allowing the inclusion of fault detection capabilities at the assembly level (graphically demonstrated in Figure 2.2-10). For example, we may have a rack of circuit boards, each of which have their own built in test capabilities. We may include an additional circuit board or piece of external test equipment whose function is to detect failures in the assembly. This additional capability will introduce its own value of FD_A and may overlap the coverage of the tests incorporated at the element levels. How do we combine the various values of FD_i with FD_A to determine FD'_+?

There are three approaches for this two-level combination. They vary in

precision and difficulty. The most precise and difficult (of course) approach is to reduce the problem to a single level by evaluating tests and their topologies and use the methods in Section 2.2.3.1. At very low

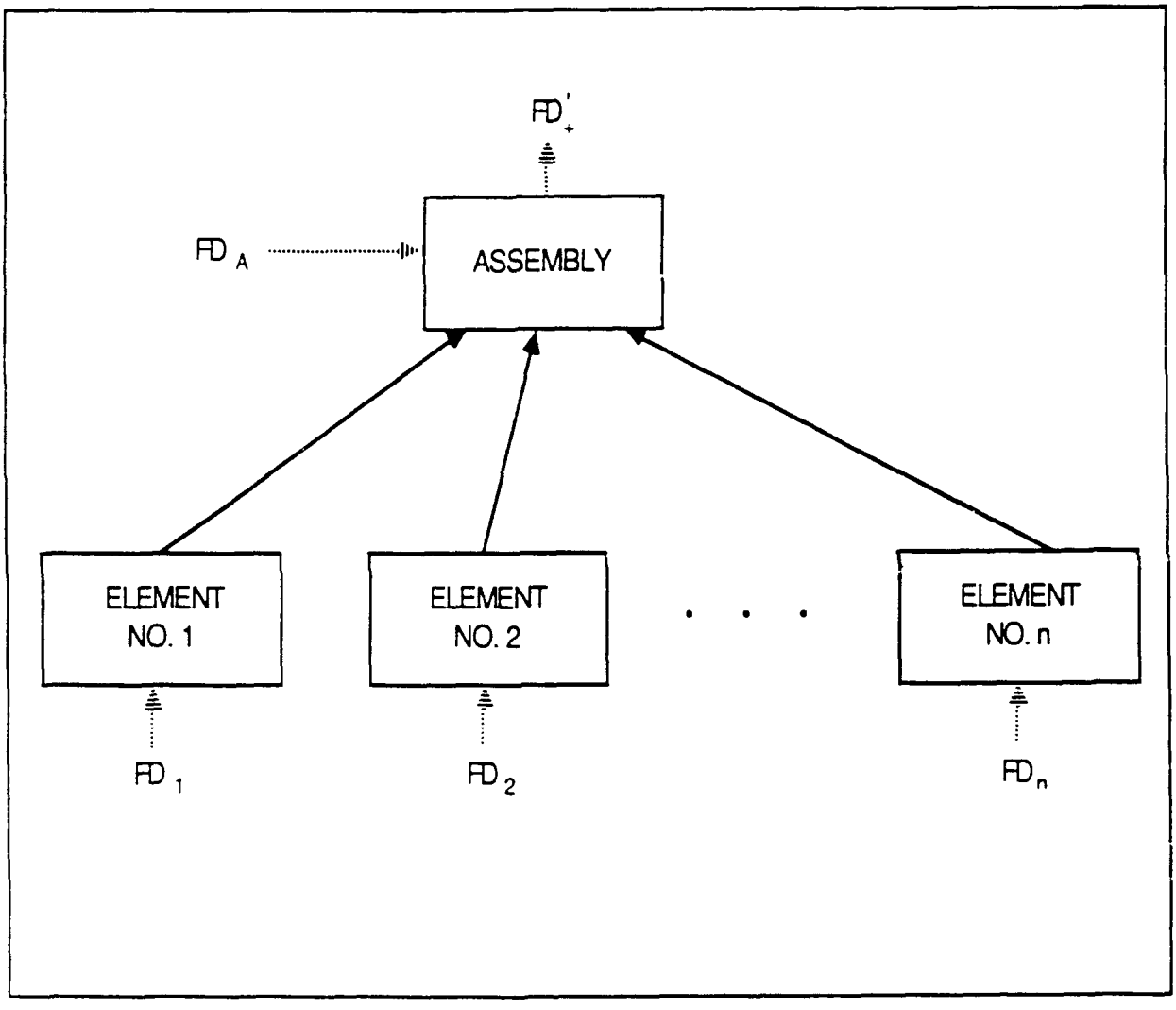


Figure 2.2-10 In addition to the fault detection capabilities incorporated in each element, (in this case FD_1) there may be additional overlapping capabilities included at the assembly level (in this example FD_A).

levels of indenture, such as component-to-circuit board, this method may be the most appropriate.

At higher levels of indenture this method becomes impractical.

The second approach is slightly less precise but constitutes less effort than the first. Given our set of elements each of which has an assessed value of FD_i and some additional detection capability at the assembly level (which may be assessed to have a value of FD_A), we decompose the coverage provided at the assembly level into its elemental values. That is to say, we find values FD_{Ai} such that

$$FD_A = \sum_{i=1}^{\text{no. of elements}} (\lambda_i / \lambda_+) FD_{Ai} \quad . \tag{2-55}$$

where FD_{Ai} is the fraction of faults in the i^{th} element that the assembly level capability detects. This concept is depicted in Figure 2.2-11.

We now make the assumption that the event wherein a fault in the i^{th} element is detected by its own tests, and the event wherein it is detected by the assembly level detection system are independent. This allows us to combine values as follows.

$$FD_i' = FD_i + FD_{Ai} - FD_i\, FD_{Ai} \tag{2-56}$$

where FD_i' is an aggregate value of fraction of faults detected for the i^{th} element due to both its internal fault detection mechanisms and those at the assembly level. For the entire assembly we now have

$$FD_+' = \sum_{i=1}^{\text{no. of elements}} (\lambda_i / \lambda_+) FD_i' \tag{2-57}$$

This approach is nice because it involves the decomposition only of those capabilities that constitute FD_A. It is substantially easier than the first method. Its major drawback is the assumption of the independence of the probabilities FD_i and FD_{Ai}. This process can be accomplished by using the approach described in Section 2.2.3.1.

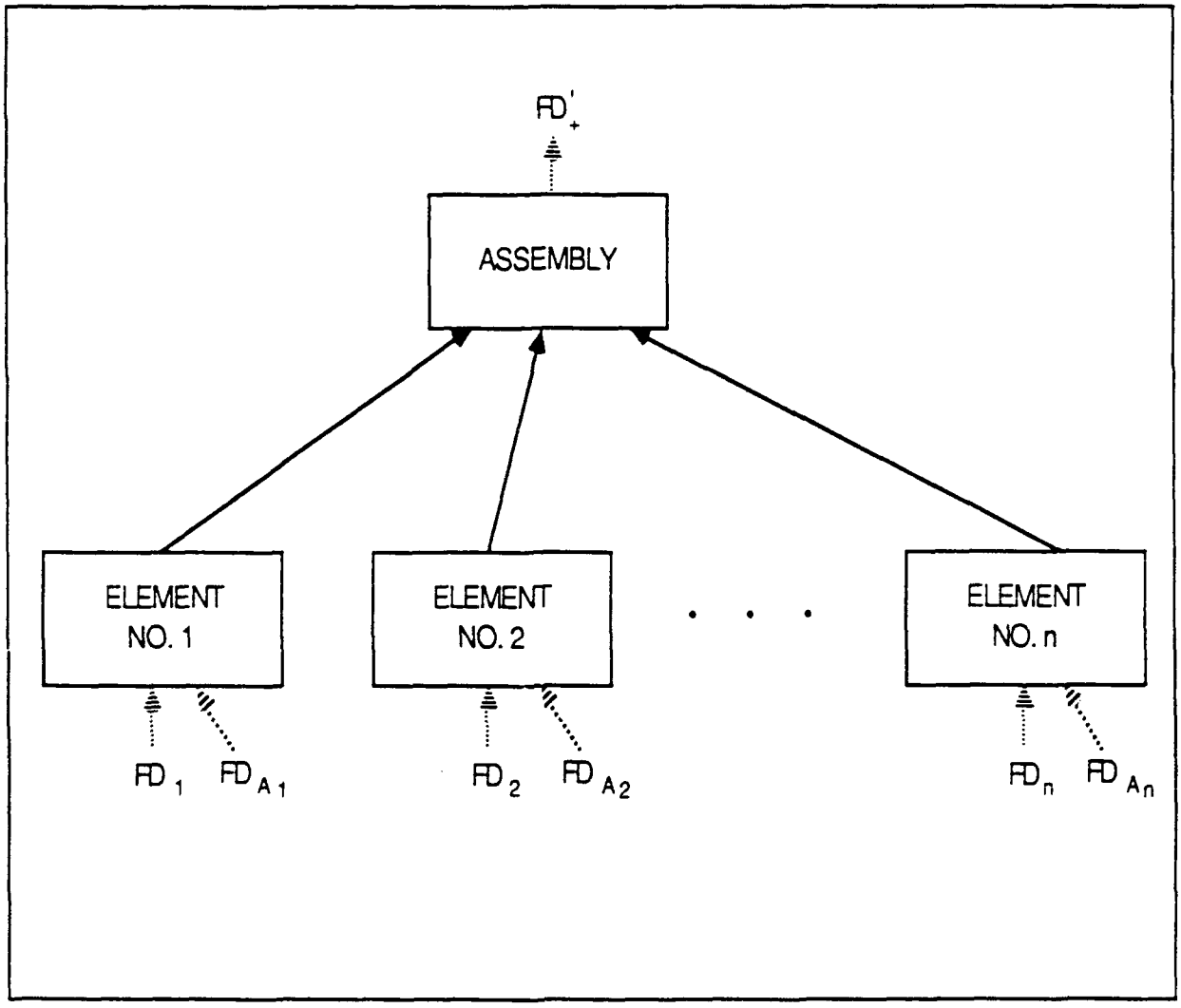


Figure 2.2-11 The assembly level contribution to testability can be reduced to equivalent contributions at the elemental level. In this case FD_A is reduced to n constituent values, FD_{Ai}.

The third approach for combining the values of FD_i and FD_A to form FD_+' is very easy and, potentially, quite inaccurate. It makes the assumption that the event of detecting a fault in any element by the elemental detection capabilities and the event of detecting a fault in any element by the assembly level capability are independent.

$$FD'_+ = FD_+ + FD_A - FD_+ FD_A \qquad (2\text{-}58)$$

where all the terms are as previously defined.

2.2.4.2 Combining FA.

The method for combining the elemental values of fraction of false alarms takes advantage of the fact that false alarm rates are additive. Thus,

$$FAR_+ = \sum_{i=1}^{\text{no. of elements}} FAR_i \qquad (2\text{-}59)$$

where FAR_i is the false alarm rate for the i^{th} element and FAR_+ is the assembly level false alarm rate. From Section 2.2.3.2 we know

$$FAR_i = \frac{FA_i}{1 - FA_i} FDR_i \qquad (2\text{-}60)$$

where FA_i is the fraction of false alarms for the i^{th} element's fault detection mechanism, and FDR_i is the fault detection rate defined in Section 2.2.4.1.

Thus, FAR_+ becomes:

$$FAR_+ = \sum_{i=1}^{\text{no. of elements}} \frac{FA_i}{1 - FA_i} FDR_i \qquad (2\text{-}61)$$

Again, from Section 2.2.3.2 we know that the assembly level value for FA_+ can be expressed in terms of the assembly values for FAR_+ and FDR_+ as follows:

$$FA_+ = \frac{FAR_+}{FAR_+ + FDR_+} \qquad (2\text{-}62)$$

These last two expressions constitute our general approach for combining the elemental values FA_i to compute FA_+.

In the case where an additional detection capability is incorporated at the assembly level, we must also compute the associated false alarm rate FAR_A.

$$FAR_A = \frac{FA_A}{1 - FA_A} \lambda_+ * FD_A \qquad (2\text{-}63)$$

The formula for aggregating false alarm rates is then modified to account for this additional term.

$$FAR_+' = FAR_+ + FAR_A \qquad (2\text{-}64)$$

Finally, the false alarm rate FAR_+' is used as before to compute the assembly level value for FA_+'.

$$FA_+' = \frac{FAR_+'}{FAR_+' + FD_+' * \lambda_+} \qquad (2\text{-}65)$$

2.2.4.3 Combining T_D

In our assembly of elements we know that the i^{th} element fails at a rate of λ_i. This value is the number of failures that are expected to occur over some time interval, say Δt. From Section 2.2.4.1 we know that we can expect a number FDR_i of fault detections (reported by the test environment under consideration) over that same time interval.

$$FDR_i = FD_i \lambda_i \qquad (2\text{-}66)$$

If the average or expected time required for detection of a fault in the i^{th} element is T_{Di}, then we can estimate the accumulated detection time, say ADT_i, spent on fault detection in the i^{th} element over the interval Δt.

$$ADT_i = FDR_i \; T_{Di} = FD_i \lambda_i \; T_{Di} \qquad (2\text{-}67)$$

Based on the same reasoning as in previous subsections, we can determine

the accumulated detection time, ADT_+, for the assembly of elements by:

$$ADT_+ = FDR_+ * T_{D+} \tag{2-68}$$

Since our failure and detection rates are additive across elements, ADT_+ can also be computed as follows:

$$ADT_+ = \sum_{i=1}^{\text{no. of elements}} ADT_i = \sum_{i=1}^{\text{no. of elements}} FDR_i \; T_{Di} = \sum_{i=1}^{\text{no. of elements}} FD_i \, \lambda_i \, T_{Di} \tag{2-69}$$

We also know from Section 2.2.4.1 that the number of detections in the time interval Δt for the assembly is

$$FDR_+ = FD_+ \, \lambda_+ = \sum_{i=1}^{\text{no. of elements}} FD_i \, \lambda_i \tag{2-70}$$

Combining both expressions for ADT_+, TD_+ is given by

$$T_{D+} = ADT_+ / FDR_+ = \sum_{i=1}^{\text{no. of elements}} (FDR_i / FDR_+) \, T_{Di} \tag{2-71}$$

Therefore the average detection time for the assembly, T_{D+}, is the assembly's accumulated detection time during the interval Δt, ADT_+, divided by the number of detections during Δt, FDR_+.
This then is a general formulation for combining multiple values of T_D across a single level of system indenture. Note, this employs a detection-rate transformation as opposed to the failure-rate transformation that was used for FD.

An approximate formulation is derived by assuming the values of FD to be very close to the average FD_+.

$$T_{D+} \approx \left(\sum_{i=1}^{\text{no. of elements}} FD_+ \ \lambda_i \ T_{Di}\right) / \left(\sum_{i=1}^{\text{no. of elements}} FD_+ \ \lambda_i\right)$$

$$= \sum_{i=1}^{\text{no. of elements}} (\lambda_i / \ \lambda_+) \ T_{Di} \qquad (2\text{-}72)$$

Note that this reduces to a simple failure rate transformation

In the more complex situation wherein we allow additional fault detection capability at the assembly level we can pursue similar strategies to those used for combining FD. There are also three approaches for this combination. As was the case with FD, the first approach is the most precise method. We reduce the problem to a single level by evaluating tests and their topologies and use the methods in Section 2.2.3.3. At sufficiently low levels of indenture this method is the most appropriate. At higher levels it becomes impractical.

The second approach decomposes the coverage provided at the assembly level into its elemental values. In Section 2.2.4.1 we sought values FD_{Ai}, the fraction of faults in the i^{th} element that the assembly level capability detects. Recall that we made the assumption that the event wherein a fault in the i^{th} element is detected by its own tests, and the event wherein it is detected by the assembly level detection system are independent. If that assumption is valid, the associated event space consists of three mutually exclusive events, each with an estimable probability.

EVENT 1: Detections are made strictly by the elemental mechanism with no participation by the assembly level mechanism.

The probability of event 1 is $FD_{1i} = FD_i \ (1 - FD_{Ai})$

EVENT 2: Detections are made strictly by the assembly level mechanism with no participation by the elemental level mechanism.

The probability of event 2 is $FD_{2i} = FD_{Ai}\ (1 - FD_i)$

EVENT 3: Detections are simultaneously made by both the elemental mechanism and the assembly level mechanism.

The probability of event 3 is $FD_{3i} = FD_i\ FD_{Ai}$

For the i^{th} element, we can expect the following accumulated detection time, ADT_i'.

$$ADT_i' = FD_{1i}\,\lambda_i\,T_{Di} + FD_{2i}\,\lambda_i\,T_{DA} + FD_{3i}\,\lambda_i\,MIN(T_{Di}, T_{DA}) \qquad (2\text{-}73)$$

where T_{DA} is the expected detection time for the assembly level capability, irrespective of the elements that are being tested. Notice that the lesser of the two times T_{Di} and T_{DA} is used for the event of overlapping fault detection coverage. It is intuitively obvious that the faster of the two detectors will report first.

The assembly level accumulated detection time, ADT_+', is calculated as before.

$$ADT_+' = \sum_{i=1}^{\text{no. of elements}} ADT_i' \qquad (2\text{-}74)$$

Similarly, T'_{D+}

$$T_{D+} = ADT'_+ / FDR'_+ \qquad (2\text{-}75)$$

Where

$$FDR_+ = \sum_{i=1}^{\text{no. of elements}} FDR_i' \tag{2-76}$$

$$FDR_i' = \lambda_i (FD_{1i} + FD_{2i} + FD_{3i}) \tag{2-77}$$

The third approach for combining the values of T_{Di} and T_{DA} to form T'_{D+} follows from the second approach. It relies on the assumption that the event of detecting a fault in any element by the elemental detection capabilities and the event of detecting a fault in any element by the assembly level capability are independent, when viewed from the assembly level perspective. As before, we decompose this event space at the assembly level into three mutually exclusive events with associated probabilities FD_1, FD_2, and FD_3. Their formulations are:

$$FD_1 = FD_+ (1 - FD_A) \tag{2-78}$$

$$FD_2 = FD_A (1 - FD_+') \tag{2-79}$$

$$FD_3 = FD_+ FD_A \tag{2-80}$$

where FD_+ is the combination of the individual elemental detection capabilities, deferred in Section 2.2.4.1. The accumulated detection time at the assembly level is the determined in a fashion similar to that used in the second method.

$$ADT'_+ = FD_1 \lambda_+ T_{D+} + FD_2 \lambda_+ T_{DA} + FD_3 \lambda_+ MIN(T_{D+}, T_{DA}) \tag{2-81}$$

where T_{D+} is the aggregate of the elemental detection times. Ultimately, the value T_{D+} is computed as the quotient of ADT'_+ and FDR'_+ where

$$FDR'_+ = \lambda_+ (FD_1 + FD_2 + FD_3) \tag{2-82}$$

2.2.4.4 Combining FI_{P+}

Whereas detection characteristics translate from lower to higher levels of system indenture, isolation characteristics have meaning at only one level. They may exist at every level, but they cannot be translated from lower to higher levels.

To understand why this is true, we examine the concept of fault isolation. At the elemental level of indenture we wish to isolate faults to the sub-element (whatever that may be). On the other hand, at the assembly level, our goal is to isolate to the element. If, in the course of performing our isolation at the assembly level, we possess elemental isolation results, then we may use that information to indict faulty elements. However, the indictment of faulty elements, when viewed at the elemental level of indenture, constitutes fault detection. In general, fault detection coverage at any level is an upper bound on the isolation coverage for the same level.

If we combine fault detection results from numerous elements and report them at the assembly level of indenture, the reports will be of the form *a fault exists in element i*. Now, if the identity of the element, *i*, is reported at the assembly level, then our combination of those results has transformed them into fault isolation results at the higher level. Thus, we use detection capabilities at one level of indenture to determine isolation characteristics at the next higher level.

The initiation of a fault isolation procedure implies that a fault has been detected by some means -- any means. If our assembly level isolation is comprised strictly of elemental detection capabilities, we compute the assembly fractional isolability, FI_{P+}, as follows. First, the assembly level value of FD_+ is determined based on the approach in Section 2.2.4.1. Since the identities of the faulty elements are being reported at the assembly level, we can take FD_+ as the probability that isolation is made to exactly one element. If not all of the elemental detection results identify the faulty elements at the assembly level, then an alternative value for FD_+ must be computed that accounts only for those detections that preserve that information. If an element fault is not covered by this translated information, then the entire assembly becomes an ambiguity group.

The denominator in the equation for computing FI_p is the expected number of sub-element removals, E{AG} (see Section 2.2.3.2).

$$E\{AG\} = FD_+ + (1 - FD_+)\ (\text{number of elements}) \qquad (2\text{-}83)$$

The term number of elements represents the number of elements that comprise the assembly.

Under the single failure assumption we have

$$FI_{P+} = 1 / E\{AG\} = 1 / [FD_+ + (1 - FD_+)\ (\text{number of elements})] \qquad (2\text{-}84)$$

This then is a general formula for combining elemental values of FD_i to derive FI_{P+}.

When fault isolation capabilities are added at the assembly level the translation becomes more complex. As is the case with FD, there are three levels of solution. Specifically they are 1) decompose the problem to a single low level (sub-elemental) wherein tests, failure modes, and their relationships are evaluated; 2) analyze the assembly-level isolation capability at a test and element level, treating the elemental detection capabilities as tests; and 3) combine the assembly level FI_{P+}, derived as shown above from the elemental values FD_i ,with FI_{PA} from the assembly level isolation capability. This last solution procedure is the same as that used to compute FD.

$$FI'_{P+} = FI_{P+} + FI_{PA} - FI_{P+}\ FI_{PA} \qquad (2\text{-}85)$$

2.2.4.5 Combining FP.

Insofar as FP is the complement of FI_P, its combination is trivial. As described above, FI'_{P+} is first determined. The following formula may be used to derive FP'_+.

$$FP'_+ = 1 - FI'_{P+} \qquad (2\text{-}86)$$

2.2.4.6 Combining T_{μ}

In the event that the assembly level fault isolation capability consists entirely of elemental detection, the expected isolation time is the same as the expected detection time, T_{D+}. As was discussed in Section 2.2.4.3 this translation is the result of a detection rate transformation. That development will not be repeated here.

In the more complex situation where fault isolation capability is provided at the assembly level, we must combine the expected time for isolation T_{D+} due to elemental detection, and the expected time to isolation for the assembly level isolation capability, T_{IA}. As discussed in section 2.2.4.4, the elemental detection capabilities can be regarded as tests from the perspective of assembly level isolation. It is not an unreasonable assumption that all such tests will be run prior to initiating isolation at the assembly level. If the fault is not detected/isolated by the elemental capability, then the assembly level isolation will be used. The probability that we will be able to detect and isolate any given fault using elemental detection is FD_{+}. Thus, our expected time to isolation is

$$T_{I+} = FD_{+} T_{D+} + (1 - FD_{+}) T_{IA} \qquad (2\text{-}87)$$

2.2.4.7 Summary

We have developed methods for translating our six TFOM's from lower to higher levels of system indenture. The figures of merit that describe the fault detection capabilities, FD, FA, and T_D, all maintain their identities as they are translated. On the other hand, isolation metrics have little meaning at higher levels of indenture and, as such, do not translate. We discovered that at a given level, I, detection capabilities at the next lower level, when translated to the given level, I , constitute isolation capabilities.

It should be noted that even though isolation capabilities do not translate, they may be aggregated across a number of units within a level of indenture. A simple failure rate transformation is used for such an aggregation. This type of combination may be necessary to demonstrate some higher level requirement.

In addition to transforming TFOM's from lower to higher levels of indenture, our methods were expanded to simultaneously combine testability metrics from two levels of indenture. For example, we can combine values of BIT FD_i from elements within an assembly, with a value FD_A provided by ETE at the assembly level, to calculate an equivalent assembly level value FD_+'.

3.0 TAM DEVELOPMENT

3.1 Introduction.

Our goal here is to develop a methodology that can be applied to cost effectively allocate testability resources across levels of indenture to satisfy some system-level performance and/or testability requirements. This testability allocation process assigns testability parameters to individual subsystems, modules, and/or LRUs to ensure the attainment of the system-level requirements.

A top-down method is presented here for apportioning the system attributes to the individual elements of a system in such a way as to optimize criterion functions. This process is the inverse of the traditional bottom-up or prediction approach. This allocation in no sense indicates that the particular level of system requirement can be achieved. It merely means that if the apportioned values are realized, the system will meet its goal or requirement.

Our allocation methodology applies to system, subsystem, LRU, and lower hierarchical system levels. This allocation is also across maintenance levels (O and D levels), and across testability resources to achieve the system testability requirements as measured by the TFOMs. These resources are BIT/BITE, ETE, Software diagnostics, Training, Technical Orders, etc.

The allocation methodology in this effort is based on optimization techniques. The aim of such techniques is to choose a set of variables such that some function of these variables, called the objective function, is maximized or minimized, subject to constraints or limitations on the variables. The objective functions and constraints must be mathematically defined. For the optimization considered here, the system performance parameters, such as Reliability (R), Maintainability, Availability (A), and Operational Readiness (P_{or}), together with Cost penalties such as Weight, Power, Volume, and cost associated with increasing levels of testability, make up the objective functions and constraints.

The organization and approach taken in the development of the TAM follows:

3.2 Problem Formulation
3.3 Literature Survey
3.4 Algorithmic Solution
3.5 Algorithm
3.6 Section Summary

3.2 Problem Formulation.

The problem of allocating testability can be viewed in two equivalent ways:

- Allocate the TFOMs (FD,FA,T_D,FI_P,FP,T_I) and/or any new developed TFOM cost effectively across the levels of indenture to satisfy system requirements;

- Determine the optimal allocation of testability resources (BIT/BITE, ETE, etc.) thereby the components TFOMs will then be determined optimally.

As stated earlier, the allocation methodology is based on an optimization technique. This optimization may be a maximization or minimization, depending on the criterion or objective function. The allocation problem can therefore be formulated as follows:

> Maximize testability given system cost/"cost functions", or equivalently minimize total cost given system testability/design requirements.

Let X_{ij} be the TFOM values to be allocated to a decomposition of a weapon system, in which the index i represents the level of indenture and j reresents the various units (subsystems, modules, LRUs) at a particular level (e.g., i=0 for major weapon system, i=1 for prime mission system, i=2 for major system, i=3 for subsystem, etc.).

Let f_{ij} be the objective function (cost/"cost functions") to be optimized, subject to a set of constraint functions C_{ij}^r (the index r represents the number of constraints) on the TFOM values.

The mathematical description is given by:

$$\textbf{MIN} \sum_{i=1}^{l} \sum_{j=1}^{n_i} f_{ij}(X_{ij}) \qquad (3\text{-}1)$$

subject to:

$$\sum_{i=1}^{l} \sum_{j=1}^{n_i} g_{ij}^{r}(X^{r}_{ij}) \leq C_r$$

$$r = 1,.....,m$$

$$0 \leq X_{ij} \leq 1$$

where:

f_{ij} & g_{ij}^{r}	:	additive and separable objective and constraints functions (no TFOM cross product)
X_{ij}	:	amount of testability or coverage to be allocated, which is bounded
l	:	number of levels of indenture
n_i	:	number of units (subsystems, LRUs) per level
m	:	number of constraints
C_r	:	maximum allowable amount of rth resource or "cost" associated with testability

3.3 Literature Survey.

Problems of form (3-1) arise in a variety of contexts, including optimal allocation of resources in search [1], the allocation of promotional resources among competing activities [2], [3], reliability [4], production [5], and subgradient optimization [6]. The allocation of a specific amount of a given resource among competing alternatives can often be modeled as a Knapsack problem, which of course, is a discrete form of (3-1).

The knapsack model formulation of the resource allocation problem, is very efficient because it allows convex cost representation with bounded variables to be solved without great computational efforts. In many instances, problem (3-1) has to be solved many times, consequently, an efficient method to solve the continous knapsack problem with bounded variables is of central interest to many applications. Moreover, a good algorithm for (3-1) may serve as a subroutine in more complex computational procedures.

Special cases and variants of problem (3-1) have been solved exactly and independently by simple, finite algorithms. These methods are based on a special property of the optimal solution called the "ranking property", in that it reflects a certain prior ranking of the variables.
Luss and Gupta in [2] subsume previous results [1], which uses convex programming arguments, and [3] which uses dynamic programming, both for particular objective functions.

They presented an iterative method mainly for strictly convex decreasing functions and a one pass algorithm for a set of particular functions with the variables bounded from below. Their method consists in relaxing the upper bound constraint, using the relaxation procedure given by Geoffrion [19]. Also, in their study they consider an inequality constraint implying that each f_j is nondecreasing at the optimum.

Zipkin [7] has extended Luss and Gupta's procedure to more general settings. His paper treats optimization problems with a nonlinear-additive objective function subject to a single linear constraint. Zipkin's algorithm can be viewed as a special form of Everett's generalized Lagrange multiplier technique [8], modified to exploit the ranking property, and in principle relaxes the equality constraint in (3-1). From a somewhat different point of view, Everett [8] drops differentiability assumptions from (3-1) and discusses a procedure for solving nonlinear programs in terms of minimizing the Lagrangian.

Bodin's algorithm [4] makes use of a ranking property for problem (3-1). Like all previous papers described above, the property derives from Karush-Kuhn-Tucker conditions [9]. This involves ranking the set of $2n_i$ numbers $\{ f'_j(o), f'_j(1), j = 1, 2, ... n_i\}$ in decreasing order. This algorithm relaxes the resource constraint while maintaining the bounds in force, while the algorithm in [2] maintains the resource constraint, the lower bounds and relaxes the upper bounds.
Bitran and Hax [5] assume only that each f_i is (not necessarily strictly) a convex function . Their algorithm applies to problems (3-1), with this assumption, and with relaxing both bounds, and solved at each iteration, that is, **one** variable is fixed at one of its bounds at each iteration. The algorithm fails when the unbounded problem (3-1) has no finite solution. Variants of problem (3-1) in the context of allocation of search effort, have also been studied by Koopman in [10], and by De Gueni [11] in the infinite -dimensional case . This case leads to results similar to the ranking property called "multiplier rules." Karush [9] gives an algorithm for general piecewise-linear f_j .

Luss and Gupta point out that their method is also applicable to the case where several resources are to be allocated, which requires repeated solution of problems of form (3-1), hence, a one pass is no longer surficient to ensure optimality. The same is true of the surrogate programing approach [12] to multi-resource problems. Shih [13] treats the case where m = 1, i. e. one resource and where x_{ij} can take on only discrete values. Mjelde [14] was first to prove that Shih's method leads to an optimal solution and Einbu [15] has shown the conditions under which such a solution is unique. Other algorithms for m > 1 are furnished by Everett [8], Danskin [16] and Mjelde [17]. It is also interesting to note that Danskin interprets multi-resource problems as finding an optimal assignment of weapons of various types to targets of various types. Einbu [18], developed a numerical method which extends the work of Luss and Gupta.

Another class of methods for solving problems of form (3-1) is the "Multiplier Methods" as discussed by Bertsekas [20]. These methods combine the Lagrangian multipliers with penalty terms, thus forming the Augmented Lagrangian Function. The main idea here is to approximate a constrained minimization problem by a problem which is considerably easier to solve. Naturally, by solving an approximate problem, we can only expect to obtain an approximate solution to the original problem. However, if we can construct a sequence of approximate problems which converges

in a well defined sense to the original problem, then hopefully the corresponding sequence of approximate solutions will yield in the limit a solution to the original problem. It may appear strange that we would prefer solving a sequence of minimization problems rather than a single problem. However, in practice only a finite number of approximate problems need to be solved in order to obtain what would be an acceptable approximate solution of the original problem. Furthemore, usually each approximate problem need not be solved itself exactly but rather only approximately. In addition, one may efficiently utilize information obtained from each approximate problem in the solution of the next approximate problem.

The analysis of the multiplier methods in terms of their convergence properties show their superiority over the ordinary penalty methods. In the pure penalty methods (i.e., Lagrangian multipliers constants or 0), it is necessary to increase the penalty parameter to infinity in order to have convergence. One advantage of the multiplier methods is the elimination or at least moderation of the ill-conditioning effects associated with large penalty parameters. A second important advantage of the method of the multipliers is that its convergence rate is considerably better than that of the penalty method. While in the method of multipliers, the rate of convergence is linear or superlinear, in the penalty method the rate of convergence is much worse and essentially depends on the rate at which the penalty parameter is increased.

A single paper on Testability Allocation Methodology [21] based on linear programming links the TFOMs to such measures as cost, mission failure probability, and hazard risk. The constraints or "cost" parameters associated with on-aircraft diagnostics burdens such as weight, volume and power as a function of the TFOM FD/FI are nonlinear, but they were linearized to expedite computations. Only one resource namely BIT was considered.

3.4 Algorithmic Solution.

The multiplier method as previously described will be used to solve the Testability Allocation problem. For simplicity we will consider one level of indenture, this will eliminate one summation from equation (3-1). It should be noted that in this case, the index i represents the unit(of which there are up to n) along the level of indenture, and j represents the particular "cost" or constraint function, of which there are up to m.

The problem is therefore equivalent to:

$$\textbf{MIN} \sum_{i=1}^{n} f_i(X_i) \qquad (3\text{-}2)$$

subject to:

$$\sum_{i=1}^{n} g_{ij}(x_i) + C_j \leq 0$$

$$j = 1,.....,m$$

$$0 \leq X_i \leq 1$$

where:

n : number of elements

m : number of constraints

The basic idea in penalty methods is to eliminate some or all of the constraints and add to the objective function a penalty term which prescribes a high cost to infeasible points. Associated with these methods is a parameter ρ, which determines the severity of the penalty and as a consequence the extent to which the resulting unconstrained problem approximates the original constrained problem. In the multiplier methods, the penalty term is added not to the objective function f but rather to the Lagrangian function L of problem (3-2) thus forming the Augmented Lagrangian function denoted by $L_\rho(x, \lambda)$ and given by:

$$L\,\rho\,(x, \lambda) = \sum_{i=1}^{n} \left[f_i(x_i) + \sum_{j=1}^{m} \lambda_{ij}\, h_j(x_i) + (\rho/2) \sum_{j=1}^{m} (h_j(x_i))^2 \right] \tag{3-3}$$

where:

$$h_j(x_i) = g_{ij}(x_i) - u_{ij}$$

The reduced optimization problem (3-2) is therefore decomposed into n scalar minimization problems, the optimal solution of each can be found by a search in the interval [0,1]. The minimization of $L\,\rho\,(x, \lambda)$ is broken down into two stages, first minimizing over all x subject to $h_j(x_i) = 0$, or $g_{ij}(x_i) = u_{ij}$, and then minimizing over all u, the subproblems resulting from the constraints of (3-2). These subproblems are defined as follows:

$$\min_{u} F(u)$$

where

$$F(u) = \sum_{i=1}^{n} (-u_{ij}\,\lambda_{ij}) + (\rho/2) \sum_{i=1}^{n} (g_{ij}(x_i) - u_{ij})^2 \tag{3-4}$$

subject to:

$$\sum_{i=1}^{n} u_{ij} + C_j \leq 0$$

$j = 1, \ldots, m$ the number of constraints

These subproblems (3-4) can be reduced to one dimensional problem by considering its dual through the reapplication of the Lagrange Multipliers, v_j. These multipliers can be interpreted as the rate of change of the objective function F(u) to the changes in testability requirements C_j. Appending the constraint to the objective function F(u) in (3-4), we obtain the Lagrangian of this subproblem:

$$\min_{u} F(u) + v_j\left(\sum_{i=1}^{n} u_{ij} + C_j\right) \tag{3-5}$$

$$v_j \geq 0$$

which can be solved explicitly for v and u.

3.5 Algorithm.

1. INPUT: Objective function ($f_i(X_i)$), constraint ($g_{ij}(X_i)$), constraints requirements (C_j), n, m which are the number of subsystems and the number of constraints respectively.

2. OUTPUT: An optimal solution, x, of problem (3-2). x represents a vector of the allocated TFOM values.

3. STEP 0: Initialize the Lagrange multipliers λ_{ij} to 0, $\lambda_{ij} = 0$ and the $u_{ij} = g_{ij}(1)$. The assumption is that the TFOM values are normalized to the range [0, 1]. Scale all the coefficients, in particular the penalty parameter, ρ .

4. STEP 1: Minimize function over x (TFOM)

For ali i compute the optimal x^*_i given by :

$$x^*_i = \min f_i(X_i) + \sum_{j=1}^{m} \lambda_{ij} g_{ij}(X_i) + (\rho/2) \sum_{j=1}^{m} (g_{ij}(X_i) - u_{ij})^2$$

The remaining steps previously described now follow:

5. STEP 2: Minimize F(u) over u

$$F(u) = \sum_{i=1}^{n} (-u_{ij} \lambda_{ij}) + (\rho/2) \sum_{i=1}^{n} (g_{ij}(X_i) - u_{ij})^2$$

6. STEP 3: Update dual variables V_j

For all j compute

$$v_j = (\rho/n) [\sum_{i=1}^{n} g_{ij}(X_i) + \sum_{i=1}^{n} (\lambda_{ij}/\rho) + C_j]$$

$$\lambda_{ij} = v_j$$

7. STEP 4: If solution is feasible (i.e. the optimal TFOM values satisfy the constraints) and converges then stop, otherwise return to step 1.

3.6 Section Summary.

The testability allocation methodology developed is general and is based on the Augmented Lagrangian method, where the objective and constraint functions could be either linear or nonlinear. In this study, we restrict ourselves to separable cost functions, that is, the objective and constraints are each functions of one TFOM, no cross products of TFOMs are allowed. The algorithm is general enough to address any number of constraints. The development and selection of meaningful objective and constraints functions follows in the next chapter.

4.0 TFOM/TAM INTEGRATION

4.1 Introduction.

The goal of the integration phase is to relate the TFOMs with viable system design parameters, to generate the objective and constraint functions for the testability optimization problem. The problem formulation as stated in the TAM development section 3.0 is two fold: First, to allocate the diagnostic parameters or TFOMs, such that the system performance (A, R, M, LCC) are met. Second, the allocation is across a mix of diagnostic resources (BIT/BITE, ETE, etc.) to achieve the testability requirements specified by the TFOMs. The testability allocation methodology is based on an optimization technique which treats diagnostic system performance as a function of the TFOMs. The purpose of this section is to include a variety of mathematical forms in order to reveal the flexibility of the solution technique. The model considered is intended as a vehicle by which to demonstrate and initiate the use of the procedure which has been developed. It is believed that the model does contain many of the significant factors which must be considered in making sound testability allocation decisions under the conditions which are specified. The data acquisition phase of generating the "cost" or objective functions and constraints resulted in three approaches:

- Analytic Relationships
- Experiential & Historical Data Bases
- Heuristic Formulation

4.2 Organization and Approach.

The section organization and corresponding approach taken in modeling the relationships of the TFOMs to the system requirements follows:

4.2.1 General Testability Model

This step describes a general testability model which will take into account the system diagnostic inadequacies such as false alarm, false isolation and failure to diagnose.

4.2.2 Measures of Effectiveness of Test Systems
The derivation of efficient measures of effectiveness at the Organizational level in terms of the system testability requirements is considered here. These "cost" parameters could be used asmeasures of allocation effectiveness.This development takesinto account the inadequacies of the test systems (i.e., false alarm, false isolation, and failure to detect/isolate).

4.2.3 Testability Influence on System Requirements
This step provides detailed relationships between the testability parameters (TFOMs) and mission/system requirements. The derivation of these relationships is via the three approaches previously mentioned, that is, analytic, experiential, and heuristic.

4.2.4 Top-Down BIT Prioritization
This step analyzes the special case where BIT is the diagnostic system or resource used to achieve the testability requirements. Graphs relating TFOMs (BIT FOMs) to cost functions are derived. In addition, relationships between design and mission parameters that involve BIT, and BIT measure of effectiveness are also derived.

4.2.5 Selection of Objective and Constraint Functions
It is shown that there are many choices for the objective and constraint functions based on the "cost" parameters derived in previous subsections. The criteria used in the selection are failure rates, mission or system performance requirements to be met, and measures of effectiveness of test systems.

4.2.1 General Testability Model.
In Avionics systems, the most important and critical level is the Organization (O) level because time is crucial and diagnostic resources at this level are limited. Consequently, automatic testing is widely implemented at this level in the form of BIT. However, the operational and

evaluation experience with BIT systems has been poor because of a high level of false alarms, CNDs, RTOKs, and false removals. One major problem in developing measures of effectiveness for diagnostic systems is the difference of interpretation for several TFOMs as was discussed in section 2.0 on TFOM development. Thus, a general testability model developed should distinguish between imperfection ' the diagnostic systems when there is a real failure and when there is no failure.

4.2.1.1 Model Structure.

At any level of repair l, the diagnostic system can be modeled as follows: A diagnostic unit which is to be tested at level l, contains n_l replaceable units (RU_i), with each RU_i containing s_i subreplaceable units (SU)($i=1,2...,n_l$). For the purpose of this study, only the O level is considered. Thus, at this level the diagnostic unit is the prime system/equipment, the RU is the Line Replaceable Unit (LRU), ($n_O=N$) subreplaceable unit is a module and the diagnostic/test system/equipment is the BIT system.

4.2.1.2 Tree Diagram.

The following testability tree diagram given in Figure 4.2-1 represents all possible testability states or events at the O-level of repair. We begin by defining the notation used for the branch probabilities in the testability tree diagram, followed by the definition of each diagnostic event in order to avoid any ambiguity that may arise.

4.2.1.2.1 Notation.

$Pr(F)_O$ Probability of prime system failure at the O level within a specified time interval.

$Pr(FD)_O$ Probability that the test system detects a fault at the O level, given that the prime system is faulty.

$Pr(FI_i)_O$ Probability that the failure is isolated to i or less LRUs at the O level, given that a fault is detected and the prime system is faulty.

$Pr(FL)_O$ Probability that all LRU's isolated at the O level are good, given that a fault is detected and the prime system is faulty.

$Pr(FA)_O$ Probability that the diagnostic system detects a failure at the O level within a specified time interval, given that the prime system is functioning properly (i.e., a false alarm).

$Pr(I|FA)_O$ Probability that any good LRU's are isolated at the O level, given that a false alarm occurred.

$Pr(EI|\ F')_O$ Probability for any good LRU that it is erroneously isolated at the O level within a specified time interval, given that the prime system is functioning properly (i.e. no failure).

$Pr(EI|F)_O$ Probability that a good LRU is erroneously isolated at the O level, given that the prime system is faulty.

$Pr(LRU_i|F)_O$ Probability that the i^{th} LRU is faulty, given that a failure exists.

N number of LRUs in the prime system.

λ_{Oi} failure rate of LRU_i at the O level.

$(FRMV)_O$ Probability of false LRU detection and/or isolation at the O level, given that the prime system is good.

$(FDG_i)_O$ Probability of failure to detect and/or isolate the failure to i or less LRU's at the O level, given that the prime system is faulty.

$(FAC)_O$ Probability of the correct action of not isolating a good LRU at the O level after reporting its failure, given that the prime system is good.

4.2.1.2.2 Diagnostic States.

The testability states or events are defined below, and the discussion is restricted to the O level. The test system can be in one of the following states:

a. Good FD/FI

The test system correctly detects and isolates the faulty LRU if a failure exists, otherwise the test system reports no failure.

b. Incorrect Isolation

The prime system is faulty and the test system detects a failure. However, the test system isolates a properly functioning LRU instead of a faulty one.

c. False Isolation

The prime system is functioning properly. However, the test system erroneously reports a failure (i.e. false alarm) and consequently a good LRU is isolated. This is measured by Pr(I|FA) and Pr(EI| F').

d. CND

The prime system is functioning properly at the O level where the test equipment reports a failure (false alarm). However, no faulty LRU is found in the isolation process.

e. Failure to detect

There is a failure in the prime system, but the test system fails to detect or report the failure.

f. Failure to isolate

The failure in the prime system is detected. However, the test system fails to isolate the failed LRU.

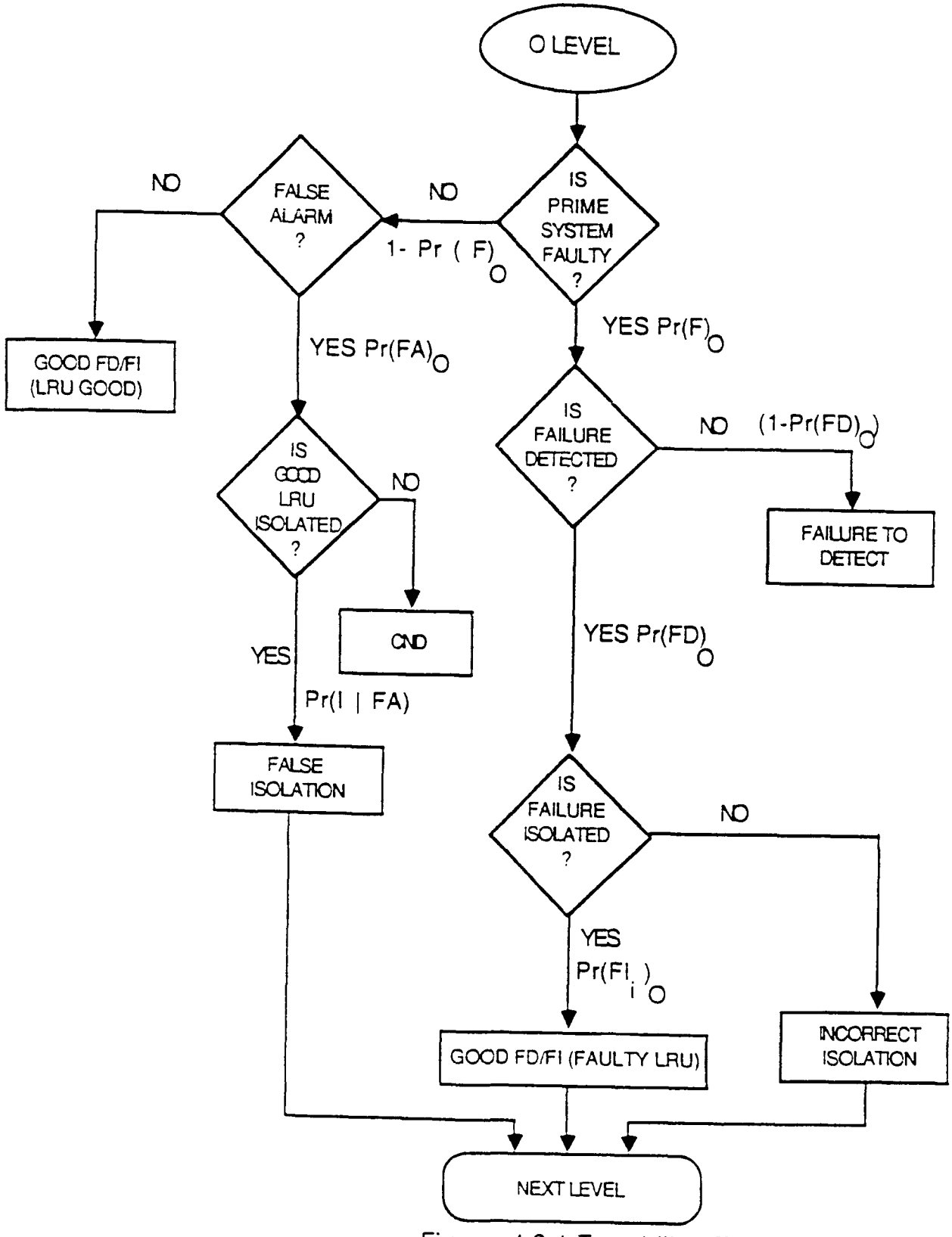


Figure 4.2-1 Testability Tree Diagram

4.2.2 Measures Of Effectiveness of Test Systems.

The central issue in testability is how effective the maintenance system is in discovering (i.e. detecting) and isolating faults. System such as automatic FD/FI which use BIT and/or ETE can be an important aid to system maintainability and system availability by reducing the need for highly skilled technicians, extensive training, technical data, and support equipment. However, in implementing automatic diagnostic systems, four types of problems can arise (as was previously described):

- False alarms
- CND
- RTOK
- Failure to diagnose

These errors reflect the inadequacies or imperfections of diagnostic systems.

The purpose of this section is to develop a methodology which can evaluate the test system capability and its relationship to system performance such as availability. In an article [22] aimed at presenting an approach to diagnostic specifications, the author points out that FD/FI systems use only the FD/FI ratio as a measure of the test system effectiveness or capability. For example, a requirement of 90% FD and 80% FI diagnostic capability means that 90% of these malfunctions addressable by the FD/FI capability are detected and of those detected, 80% are isolated. However, since the fraction of faults detected and isolated are independent in the statistical sense, then 72% (90% x 80%) of the addressable malfunctions can be isolated. The author points out that there are constraints (CND, RTOK) that would degrade the automatic diagnostic capability in relation to stated requirements. Thus, the FD/FI figure is misleading, since it does not take into account the undetected faults. Moreover, this figure is ambiguous with respect to how false alarms, false isolation and CND are to be interpreted. The goal is therefore to use system requirements to derive efficient measure of effectiveness at the 0-level taking into account the imperfections of the test systems. The section begins by the assumptions applicable to the model, followed by the computation of the testability parameters using the testability tree diagram of Figure 4.2-1. The derivation of measures of effectiveness in terms of these testability parameters follow.

4.2.2.1 Assumptions.

The following conditions and assumptions are applicable to the model which follows:

a. The test system can be considered as consisting of two processes:

- A detection process indicating a failure somewhere in the prime system.
- An isolation process which is only realized after a failure is detected. This process depends on the maintenance strategy used to isolate the failed unit.

b. Only one LRU can fail at any time, that is, a single failure assumption.

c. Detection and Isolation processes at the O level do not cause the prime system to fail.

d. The probability of erroneously isolating a good LRU is equal in all LRUs.

e. Erroneous Isolation events can occur independently in all good LRUs.

f. False Isolation events as a result of false alarm can also occur independently for all good LRUs.

g. All LRUs have a constant failure rate. Such rates correspond to the LRUs in their mature, useful state, after the periods of transients and infant mortality.

4.2.2.2 Computation of Testability Parameters

An analysis of the test system capability is given by computing the testability parameters using the tree diagram. As was stated in [22], the FD system at any level should take into account both addressable and non-addressable faults.

$Pr(fLRU_i)_O$ = Probability that the diagnostic system will detect a fault in LRU_i at the O level, given that LRU_i is faulty.

$Q(ad_i)_O$ = Percentage of all addressable faults in LRU_i which can be detected by the diagnostic system.

$Q(f)_O$ = Percentage of all possible faults in LRU_i which are addressable by the diagnostic system.

The equations given below are derived using the testability tree diagram (figure 4.2-1)and the laws of probability. (Note, the subscript O refers to the O level.)

4.2.2.2.1 Probability of Fault Detection

$$Pr(FD)_O = \sum_{i=i}^{N} Pr(LRU_i|\ F)_O * Pr(fLRU_i)_O \qquad (4\text{-}1)$$

where

$$Pr(LRU_i|\ F)_O = (\lambda_{Oi}) / \sum_{i=1}^{N} \lambda_{Oi} \qquad (4\text{-}2)$$

$$Pr(fLRU_i)_O = Q(ad_i)_O * Q(f)_O \qquad (4\text{-}3)$$

EQUATION 4-4 WITHDRAWN

4.2.2.2.2 Probability of False Isolation.

False isolation can occur as a result of the false alarm (measured by $Pr(FA)_0$). As previously stated, the Pr(I|FA) and Pr(EI|F') can be used to measure the false isolation. Given that these events can occur independently in the N LRUs (assumption f), the probability of false isolation of any good LRU's given that a false alarm has occurred, is given by:

$$Pr(I|FA)_0 = 1 - [1 - Pr(EI|F')]^N \qquad (4\text{-}5)$$

4.2.2.2.3 Probability of CND.

The CND event, as can be seen in the testability tree diagram, can be measured by the probability of isolating no LRU given that there is no false alarm after subsequent troubleshooting. (Real failures which are not isolatable are classified under "Failure To Diagnose")

$$Pr\ (CND)_O = (N)!\ /[(0)!\ *\ (N)!]\ *\ [Pr(EI|F')_0]^0\ *\ [\ 1-\ Pr(EI|F')_0]^{(N)}$$

$$Pr\ (CND)_O = [\ 1-\ Pr(EI|F')_0]^{(N)} \qquad (4\text{-}6)$$

4.2.2.2.4 Probability of Prime System Failure.

Assuming the LRUs are in series and they have constant failure rate (assumptions a and g) we have:

$$Pr(F)_O = 1 - \prod_{i=1}^{N} \exp(- \lambda_{Oi} * T_m) \qquad (4\text{-}7)$$

where T_m is mission time as defined in section 1.1.1.

4.2.2.2.5 Average Ambiguity Level.

The $Pr(FI_i)_O$ is a measure of the correct isolation capability of a test system. For computational purposes, isolation requirements are combined into a single measure of the test system's fault isolation capability. This is accomplished by computing an expected value for the fault isolation requirements. The value derived will be called the average ambiguity level E(FI) given by:

$$E(FI) = \sum_{j=0}^{N} [Pr(FI_j) - Pr(FI_{j-1})] * j$$

$$Pr(FI_o) = 0 \qquad (4\text{-}8)$$

4.2.2.3 Computation of the Measures of Effectiveness.

This section defines four measures of effectiveness of test systems derived from the testability system requirements. They are, False removal (FRMV), Failure to diagnose (FDG), False alarm correction (FAC) and Expected number of removals per failure, E(RMV).

4.2.2.3.1 False Removal.

At the O level, if there is no failure in the prime system, then the test system (BIT, ETE, etc.) should not detect or isolate any (properly functioning) LRU. If that happens, then the test system commits an error which causes a false alarm. This measure, FRMV, is computed as the probability of falsely detecting and isolating the LRU at the O level given that there is no malfunction in the prime system over a specified time interval. It is a function of false alarm and false isolation. Thus, FRMV represents a path in the testability tree diagram and is given by

$$(FRMV)_O = [1 - Pr(F)_O] * Pr(FA)_O * Pr(I|FA)_O \qquad (4\text{-}9)$$

4.2.2.3.2 Failure to Diagnose

At the O level, if there is a failure in the prime system then the test system should detect and isolate the faulty LRU. A failure to diagnose error occurs if a fault occurs and the test system eitner fails to isolate to the prescribed ambiguity 'i' or it fails to report the faulty LRU, or it isolates to only good LRU's instead. The FDG measure is represented by the probability of failure to detect and /or isolate the faulty LRU at the O level, given that the prime system has malfunctioned. It is a function of the FD/FI capability of the system, and represents the capability of correct diagnosis. Using the tree diagram, FDG is computed as follows:

$(FDG_i)_O$ = [Probability of failure to report]

+ [Probability of failure to isolate the LRU]

+ [Probability of incorrect LRU isolation]

$$(FDG_i)_O = [\, Pr(F)_O - Pr(F)_O\, Pr(FD)_O \,]$$

$$+ [\, Pr(F)_O\, Pr(FD)_O - Pr(F)_O\, Pr(FD)_O\, Pr\,(FL)_O - Pr(F)_O\, Pr(FD)_O\, Pr(FI_i)_O$$

$$+ [\, Pr(F)_O\, Pr(FD)_O\, Pr\,(FL)_O$$

$$(FDG_i)_O = Pr(F)_O\, [1 - Pr(FD)_O\, Pr(FI_i)_O] \qquad (4\text{-}10)$$

The FRMV and FDG measures represent the accuracy of the test system and its ability to perform according to the requirements.

4.2.2.3.3 False Alarm Correction.

The false alarm correction, FAC, is defined as the ability of the test system to correct its actions after erroneously detecting or isolating a failure. If the prime system is not faulty, the test system should not report a failure, or isolate a good LRU. Thus, if no LRU is isolated after erroneously reporting a failure either at the same level, or at any subsequent levels, then this measure, to some extent, eliminates the effect of a wrong decision. This measure is simply a function of the CNDs and the RTOKs at different levels. It is then the probability of the correct action of not isolating a properly functioning LRU at the O level after reporting its failure, given that the prime system is functioning properly. This measure represents the ability of different test systems at the same level or even at different levels to correct its actions after a false alarm. Using the tree diagram, we get

$$(FAC)_O = 1 - Pr(I|FA)_O \qquad (4\text{-}11)$$

4.2.2.3.4 Expected Number of Removals per Failure

Diagnostic systems, as previously stated, are used as a means to detect and isolate prime system failures. Isolation of a failure to one unit or LRU is ideal in the sense that it will result in only one removal per failure (assuming 0% false alarm). In addition, maintenance resources spent to return a prime system to working order will be minimum. Thus, the expected number of removals per failure, E(RMV) can be used as a measure

of the test system capability to perform its designated task. It is shown that this measure is also derived from the system requirements. We begin by computing the removal rate.

a. Computation of the Removal Rate.

The term RMV will be used to define the number of LRUs removed, a rate which is dictated by the isolation process. The calculation of the removal rate and hence of the E(RMV) is based on the following assumptions.

- One single failure within the system (assumption b)
- Fault indicated LRUs are removed, replaced, and retested on a random individual basis until the faulty LRUs are located.
- FD capability of test systems is equal in all LRUs.
- The probability of erroneous isolations is equal in all LRUs (assumption d)

Three cases are considered in deriving the average number of removals:

Case 1:

In the case where fault detection occurs via standard maintenance, the isolation process is initiated and we can say that slightly over one half of the fault indicated LRUs will be removed before the failed LRU is found. Let,

n be the number of erroneous isolations of diagnostic groups.

E(FI) be the expected value for the fault isolation requirements,or the average ambiguity level as defined previously

RMVFD the average number of removals given that fault detection occurs

then

$$RMVFD = (1/2) * [(n + 1) * E(FI) + 1] \quad (4\text{-}12)$$

Case 2:

If no fault detection occurs by standard means but failure exists and the isolation process may be initiated by the pilot or by other means, either no information or incorrect information is given concerning the failure location. This will lead to first removing, replacing and retesting the failure indicated LRUs, which is n * E(FI) where n is defined as previously. Given that the failed LRU is not within this group, and if N is the number of the LRUs in the system, then the [(N - n) * E(FI) + 1] /2 remaining LRUs will then be removed. Thus, the average number of removals in this case denoted by RMVNFD is given by

$$RMVNFD = (1/2) * [(N + n) * E(FI) + 1] \quad (4\text{-}13)$$

Case 3:

Let us assume that when false alarm occurs, a good spare or retesting will remove the factors causing this false alarm. Hence, the minimum a false alarm occurrence will lead to is a maintenance action, and possibly a removal. Thus, the average number of removals due to false alarm and denoted by RMVFA is conservatively estimated as:

$$RMVFA = 1. \quad (4\text{-}14)$$

This formula states also that it is up to the user to come up with the best estimate of the number of LRUs removed due to false alarm, i.e., RMVFA, taking into account the maintenance strategy.
A general formula for RMVFA will be numbers between 0 and E(FI), that is,

$$0 \leq RMVFA \leq E(FI). \quad (4\text{-}15)$$

b. Computation of E(RMV).

The value of E(RMV) is based on the assumptions stated previously, that is, a single failure within the system, and an iterative removal policy. The total E(RMV) takes into account detection and isolation by standard maintenance , other means, and n erroneous isolations which occur in both cases. Thus, given a set of test system requirements, E(RMV) is given by

$$E(RMV) = E(RMV)_{FD} + E(RMV)_{NFD} + E(RMV)_{FA} \quad (4\text{-}16)$$

where:

$E(RMV)_{FD}$ is the expected number of removals per failure due to fault detection and isolation by standard maintenance.

$E(RMV)_{NFD}$ is the expected number of removals per failure where isolation is done by other means.

$E(RMV)_{FA}$ is the expected number of removals per failure due to false alarm.

These values are determined as follows:

$$E(RMV)_{FD} = Pr(FD)_O * RMVFD \quad (4\text{-}17)$$

$$E(RMV)_{NFD} = (1 - Pr(FD)_O) * RMVNFD \quad (4\text{-}18)$$

$$E(RMV)_{FA} = \lambda_{FA} * RMVFA \quad (4\text{-}19)$$

where:

λ_{FA} is the number of false alarm occurrences per system failure.

Note: The values of FRMV, FDG, and FAC are between 0 and 1. Smaller values of FRMV, FDG and E(RMV) indicate that we have more effective test systems, while a smaller value of FAC indicates an inferior test system capability.

4.2.3 Testability Influence on System Requirements.

In the selection and development of TFOMs as stated in the introduction of this report, section 1.1.4, one of the criteria is that the TFOMs must be relatable to mission/ system parameters of maintainability, reliability (R), operational readiness (P_{or}), and cost (LCC), etc. The purpose of this section is to develop the relationships between the testability requirements or TFOMs and the above mentioned system parameters. These parameters are considered system requirements which are determined by prime system operational analysis. The derivation of these relationships are via the three approaches described previously, that is, analytically, experientially, and heuristically. The description of the three approaches follows:

- The Analytical relationships between testability parameters and system performance such as R, MTTR, A, and LCC are derived.

- MIL-HDBK 217 provides Experiential and Historical Data Bases for calculation of Failure-rate weighted coefficients for the objective functions and constraints. MIL-HDBK 472 provides repair time subtasks data used in the maintainability calculations. The MATE Guide (G3V3P2, Appendix E dated 1 April 1985) provides the data and procedures for the calculations and generation of graphs of the testability burden factors or cost functions relating TFOMs to the different ways of accomplishing diagnostics.

- The Heuristic (intuitional) approach provides experiential inferences such as, "cost" functions vs TFOMs (e. g., Design cost vs Fault Detection percentage). The linearization of relationships between TFOMs and "Costs"(i. e., weight, volume, power, etc.) is also based on heuristics.

The "cost" parameters considered in this study may be classified as follows:

- Testability requirement
- Maintenance requirement
- Operational readiness requirement
- Mission reliability requirement
- LCC requirement
- Maintenance manpower requirement
- Overhead burden requirements

A discussion of each "cost" appears in the sections which follow.

4.2.3.1 Testability Requirement

This requirement arises from the following considerations:

1. Upper and lower bounds for testability requirements or TFOMs should be included as a constraint in calculations. The maximum limit for the TFOMs for each subsystem is 100%. This constraint takes the following form:

$$lb_i \leq TFOM_i \leq ub_i. \tag{4-20}$$

2. The system consisting of N subsystems should be designed with at least a coverage factor or testability requirement (TFOM) of C (i. e. system-level coverage) so that the testability constraint is of the form:

$$\sum_{i=1}^{N} \left(\lambda_{Oi} / \lambda_S \right) TFOM_i \geq C \tag{4-21}$$

where:

λ_{Oi} = failure rate of the i^{th} subsystem

λ_S = total failure rate of the system, i. e. $\sum_{i=1}^{N} \lambda_{Oi}$

$TFOM_i$ = coverage factors for i^{th} subsystem to be allocated.

4.2.3.2 Maintenance Requirement

The maintenance or time limitations constraints are the diagnostic system's capability to meet the restrictions imposed by the operational and maintenance concepts. The index determining this quality is system mean-time-to-repair (MTTR). The time required to perform maintenance on a system may be divided into four separate tasks times as described in section 1.1.1 as follows:

$$T_M = T_P + T_{FI} + T_{RR} + T_C \qquad (4\text{-}22)$$

where:

T_M = time to perform maintenance

T_P = time to prepare system for maintenance

T_{FI} = time to isolate the failed item (subsystem, module, LRU)

T_{RR} = time to remove and replace the failed item

T_C = time to checkout the system after maintenance has been performed.

In general, the times required to perform each of the tasks noted above are independent random variables which obey certain underlying distribution functions. Under these conditions MTTR is given by:

$$MTTR = E(T_M) = E(T_P + T_{FI} + T_{RR} + T_C) \quad (4\text{-}23)$$

where $E(T_M)$ is the expected value of T_M. The MTTR constraint becomes:

$$E(T_P + T_{FI} + T_{RR} + T_C) \leq MTTR_{MAX} \quad (4\text{-}24)$$

where

$MTTR_{MAX}$ = maximum allowabe MTTR.

From the laws of probability, the above equation may be rewritten as follows:

$$E(T_P) + E(T_{FI}) + E(T_{RR}) + E(T_C) \leq MTTR_{MAX} \quad (4\text{-}25)$$

The following assumptions are used in the determination of the terms in equation (4-25):

(1) Failure of the system/equipment results from the failure of a single item (subsystem, module, LRU) within the system.

(2) Corrective maintenance action consists of:

- preparing the system/equipment for fault isolation.
- fault isolating the failed item.
- obtaining a spare of the failed item from the inventory and replacing it in the failed system.
- performing standard checkout procedure on the maintained equipment to insure that it is operational.

(3) A spare item is always available to replace a failed item (no backorders ever occur)

(4) Fault isolation of the items in a failed system/equipment is

accomplished by testing the individual items in a standard recommended sequence as determined by the design engineers. However, in this report as was stated in the computation of the removal rate, the isolation strategy is accomplished in a random sequence.

(5) The time required to checkout a specific item is exponentially related to the item complexity. In the current study, the number of parts or the failure rate may be selected as an index of item complexity.

(6) The distribution of the random variables T_P , T_{RR} , T_C are independent of the modular configuration of the system/equipment.

We will also assume that the time to fault isolate the item, T_{FI}, is also exponentially related to the item complexity (see assumption 5). The functional relationship taken is as follows:

$$T_{Ci} = T_{Pi} + a * \exp(b * \text{complexity of item i}) \qquad (4\text{-}26)$$

where:

T_{Ci} Time required to checkout item i

T_{Pi} Time required to prepare item i for checkout. This time is assumed to be constant, independent of item complexity.

a, b model constants

In this study, to derive an upperbound for MTTR, we assume that the testing of the failed items is done in a random fashion as stated in assumption 4. The reason for this assumption follows:

• The top-down allocation process is conducted early in the design phase of the system acquisition process and it is therefore very unlikely that a fault symptom matrix has as yet been developed. A symptom matrix would necessarily be dependent on the final design configuration and would have a definite effect on the recommended sequence for item fault testing.

• The random selection sequence yields an overestimate of the required fault isolation time.

Thus, in view of the above considerations, a random selection sequence of the items will yield reasonable estimates of MTTR.

The expected value for T_{FI} is given by:

$$E(T_{FI}) = \sum_{i=1}^{N} Pr(C_i) * T_{Ci} \qquad (4\text{-}27)$$

where

C_i event of selecting item i in any given fault isolation testing sequence.

Since item i is either failed or not failed, $Pr(C_i)$ for each item may be computed as follows:

$$Pr(C_i) = Pr(C_i | F_i) * Pr(F_i) + Pr(C_i | F'_i) * Pr(F'_i) \qquad (4\text{-}28)$$

where

$Pr(F_i)$ probability that item i has failed

$Pr(F'_i)$ probability that item i has not failed

$Pr(C_i | F_i)$ conditional probability that item i is chosen in the test sequence given that it has failed

$Pr(C_i | F'_i)$ conditional probability that item i is chosen for testing given that item i has not failed.

Since the random selection of items for fault checkout will not terminate until the failed item is isolated, one gets the relation:

$$Pr(C_i|F_i) = 1. \quad (4\text{-}29)$$

and

$$Pr(C_i|F'_i) = d \quad (4\text{-}30)$$

where d is a constant.

The probability of failure of any given item can be validly approximated by:

$$Pr(F_i) = \lambda_{Oi}/\lambda_S \quad (4\text{-}31)$$

and

$$Pr(F'_i) = 1 - \lambda_{Oi}/\lambda_S \quad (4\text{-}32)$$

where the above terms are as previously defined.

Combining the above results, the equation for $Pr(C_i)$ becomes::

$$Pr(C_i) = (1) * \lambda_{Oi}/\lambda_S + (d) * (1 - \lambda_{Oi}/\lambda_S) \quad (4\text{-}33)$$

Rearranging equation (4-33), equation (4-27) then becomes:

$$E(T_{FI}) = \sum_{i=1}^{N} [d + (1-d) * (\lambda_{Oi}/\lambda_S)] * T_{Ci} \quad (4\text{-}34)$$

where T_{Ci} for any given item is obtained from equation (4-26).
Estimates for values of $E(T_P)$, $E(T_{RR})$ and $E(T_C)$ should be provided from the environment in which the system is to be placed, and taken as constants using MIL-HDBK 472.

Therefore, equation (4-25) becomes:

$$E(T_M) = T_1 + \sum_{i=1}^{N} [\, d + (1-d) * (\lambda_{Oi} / \lambda_S)\,] * T_{Ci} + T_2 + T_3 \qquad (4\text{-}35)$$

where:

$$T_{Ci} = T_{Pi} + a * \exp(\, b * \text{complexity of item i})$$

T_1, T_2, T_3 , are the expected values of $E(T_p)$,$E(T_{RR})$ and E(Tc) respectively.

4.2.3.3 Operational Readiness Requirement.

Although operational readiness is primarily a requirement imposed on the overall weapon system, it is a definite consideration in the design of embedded test systems (e. g. BIT). Operational readiness is a complex function of equipment reliability, corrective maintenance time, mission time, duty cycle, functional performance threshold, and failure detectability. For the purpose of quantification, P_{or} can be expressed as:

$$P_{or} = R(\, T_m\,) + Pr(D) * Pr(\, T_r \leq T_c\,) * (1 - R(\, T_m\,)) \qquad (4\text{-}36)$$

where

R(Tm)	Previous Mission Reliability
T_r	Time required to effect repair, or repair time for which maintainability is estimated
T_c	Checkout time, or fixed time between missions
Tm	Time to complete the mission

all the other terms, Detectability (Pr(D)), and Pr(Tr $\leq$ Tc) are actually system parameters which are defined in section 1.1.3.

It is clear that the basic factors in the determination of P_{or} are reliability

particular level of operational readiness is entirely dependent upon these three variables. Of the three, reliability is the least likely to be improved to any appreciable extent.

Maintainability is actually affected by two areas of influence - Weapon system design and test system design. Such weapon system features as LRU accessibility, weight, volume and complexity can be controlled positively through careful planning and coordination but generally are limited as candidates for improving operational readiness because of operational considerations. However, test system design can substantially impact maintainability through two means. The first is through improved fault isolation which reduces the time necessary for troubleshooting. The second is through improved and more thorough testing which not only speeds up reverification testing, but reduces repetitive maintenance resulting from removal and replacement of LRU's erroneously identified as failed. Thus, the fault isolation capability of the test system weighs heavily in determining whether or not a weapon system can meet a particular operational readiness goal.

The primary aspect of the test system, fault detection capability, is the third basic factor influencing P_{or} directly. As can be seen, the greater the proportion of failures detected immediately the sooner the problem can be addressed and the sooner corrective maintenance can be accomplished.

Time is also the all-important factor in effective maintenance since an overall goal is the reduction of the weapon system downtime. Consequently, the embedded test system must contribute to time savings in those maintenance tasks where testing is involved. Thus, the basic task is the reduction of time in detecting failures and in locating those failures without sacrificing accuracy or completeness of testing. If only the correction of failures is considered in the maintenance tasks, ignoring administrative time and replacement of expendables, M_{ct}, mean corrective time can be obtained by using activity categories and associated times as given in MIL-HDBK 472.

Maintainability can be expressed using the exponential approximation:

$$Pr(T_r \leq T_c) = 1 - \exp(-T_c / M_{ct}) \quad (4\text{-}37)$$

Define the Isolation certainty, I, as the probability (expressed as a percentage) that the test system can correctly ascertain and indicate which LRU must be removed and replaced to correct a particular detected failure.

Define T_I as the fault isolation time associated with the isolation certainty I of the test system, then for I = 100%, $T_I = 0$.
Hence, for any I between 0% and 100 %,

$$T_I = [\, M_{ct0} - M_{ct1} \,] (1-(I/100)) \quad (4\text{-}38)$$

where:

$$M_{ct0} = M_c, \; (I = 0\%), \text{ and } M_{ct1} = M_{ct} \; (I = 100\%)$$

Thus, the total corrective time would be increased by this increased isolation time, that is,

$$M_{ct} = M_{ct1} + T_I \quad (4\text{-}39)$$

The equation for maintainability becomes:

$$Pr(T_r \leq T_c) = 1 - \exp[\, - \, T_c / (M_{ct1} + T_I) \,] \quad (4\text{-}40)$$

In the derivation of P_{or}, the mathematical treatments assume independent failures, with no partial degradation, no intermittent phenomena and no incorporation of false alarms as relevant system failures. In the above equation $R(T_m)$, the reliability at T_m ; i.e., the probability that the system did not fail during the interval $[0, T_m]$, is treated as independent of BIT and thus P_{or} is monotonic and bounded by $R(T_m)$ and 1. However, even if we allow the first two assumptions that BIT will not degrade reliability, and will accurately detect all failures it is designed to detect,

the addition of false alarms considerations will clearly modify the operation readiness equation. Malcolm and Highland [23], in their study of BIT false alarms correctly determined that "BIT false alarms should be considered a top contributor to the problem of excessive support costs for fielded military electronic systems."
None of the literature reviewed developed a mathematical model which relates BIT to the problem of false alarms.
The following equation describes the impact of false alarms on operational readiness. If we define T_{FA} as the time needed to detect that a failure indication is actually a false alarm, and Pr(FA) as the probability that BIT falsely indicates a malfunction during a mission, then P_{or} becomes:

$$P_{or} = \{ (1 - Pr(FA)) + [Pr(FA) * Pr(T_{FA} \leq T_c | \text{ false alarm })]\} \; R(T_m) + [Pr(D) * Pr(T_r \leq T_c) * Q(T_m)] \qquad (4\text{-}41)$$

It is clear that high probabilities of false alarms coupled with the imperfections of O level false-alarm-detection capabilities can make P_{or} non-monotonic. Any degradations in reliability due to the test system BIT will only aggravate this situation.

Note:
In (4-36), the equality is used although operational readiness is actually a level to be achieved by the system and that level may be exceeded.
On the assumption that operational readiness, mission reliability, scheduled time between missions and maintainability are known or given, it is possible to rewrite equation (4-36) as:

$$Pr(D) = (P_{or} - R(T_m)) / (Q(T_m) * Pr(T_r \leq T_c)) \qquad (4\text{-}42)$$

A bound on detectability (testability parameter) is provided by using equation (4-42).

4.2.3.4 Mission Reliability Requirement.

In the equation for P_{or} , $R(T_m)$ is the mission reliability or probability of previous mission success. This can be expressed using the exponential approximation:

$$R(T_m) = \exp(- T_m * \lambda_m) \quad (4\text{-}43)$$

where λ_m is the mission failure rate :

$$\lambda_m = \lambda_S + \lambda_{BIT} \quad (4\text{-}44)$$

and λ_{BIT} is the failure rate due to BIT circuitry.

4.2.3.5 LCC Requirement.

The purpose of a cost analysis is, of course, to determine the most cost-effective testability allocation policy. Life cycle costs are subdivided into:

- Research and Development
- Acquisition Costs (which includes Design, Production and Initial support cost)
- Operation and Support (O & S) Costs

This involves a somewhat detailed comparison of development costs, production costs, and support costs of all of the candidate test system configurations, including the cost of any external test equipment necessary to supplement the embedded test system in achieving full failure detection and isolation capabilities. The question is what is the relationship between embedded test systems or ETE effectiveness and life cycle costs? Obviously, because of diverse system missions, environment, and support concepts, there is no unique solution to this question.
The approach taken in analyzing the influence of the test systems upon the life cycle cost elements is divided into the following:

4.2.3.5.1 Cost of embedded test systems

4.2.3.5.2 Costs associated with the measures of effectiveness

4.2.3.5.3 Life cycle cost

4.2.3.5.1 Cost of Embedded Test Systems.

The cost model recommended for use in determining cost-effectiveness of the embedded test system is the one given in MIL-STD-1591 (On-Aircraft, Fault Diagnosis, Subsystems, Analysis/Synthesis Of). The cost elements can be identified and more simply defined as follows:

1. Cost of Development.
2. Average cost of production unit.
3. Cost of auxiliary equipment to support or complete the testing.
4. Cost of maintaining all the required auxiliary test or maintenance equipment.
5. Cost of manhours for failure detection (detectable failures).
6. Cost of manhours for failure isolation (detectable failures).
7. Cost of manhours for failure detection (nondetectable failures).
8. Cost of manhours for failure isolation (nondetectable failures).
9. Cost of maintaining embedded test systems.
10. Cost of preventive maintenance on embedded test systems.

The cost equation is shown below:

$$\text{Cost} = C_D + NC_P + C_{aux} + ZC_{maux}$$

$$+ (1 - P_F) [N_F \lambda_{PE} T Z (MMH_i + MMH_s)](C_{MH})$$

$$+ P_F (N_F \lambda_{PE} T Z) [(MMH_{RP}) (C_{MH}) + C_{FD}] \qquad (4\text{-}45)$$

$$+ N_F \lambda_I T Z [C_{IFMA} + (C_{IFMP})(C_{MH})]$$

$$+ (N_F T Z / T_{PM}) (MMH_{PM}) (C_{MH})$$

where:

C_D Development cost of the embedded test system.

N number of units of embedded test systems or units containing embedded test systems.

C_P Average production cost of embedded test system(the average cost of a single unit.)

C_{aux} Total cost of any auxiliary test or maintenance equipment, external to embedded test system, required to support or complete fundamental embedded test system tasks. (For example, a supplemental piece of test equipment necessary to complete a fault isolation task.)

Z Number of years the embedded test system is contemplated to be in service.

C_{maux} Cost per year of maintaining all required auxiliary test or maintenance equipment.

P_F Proportion of prime equipment's faults not detected by applicable embedded test systems.

N_F	Average number of units of embedded test systems or units containing test systems in field use at any time.
λ_{PE}	Failure rate of prime equipment(s) which the embedded test system serves (does not include failure rate of parts belonging uniquely to the test system), in failures/flying hour.
T	Flight hours/unit embedded test system/year
MMH_i	Average maintenance manhours required for initial fault detection and isolation by the embedded test system (NOTE: If fault detection and isolation is fully automatic, $MMH_i = 0$).
MMH_S	Average maintenance manhours required for secondary isolation (to detemine which is the malfunctioning LRU in those cases where initial isolation is ambiguous).
C_{MH}	Cost per maintenance manhour.
MMH_{RP}	Average maintenance manhours required for manual troubleshooting to isolate to an LRU in those cases when a failure is not detected by the embedded test system.
C_{FD}	Average cost to determine that a failure has occurred.
λ_I	Failure rate of the embedded test system.
C_{IFMA}	Average cost per embedded test system failure (material, spares, etc.).
C_{IFMP}	Average number of manhours required to repair an embedded test system failure.
T_{PM}	Flight hours between preventive maintenance actions for embedded test system.

MMH_{PM} Average maintenance manhours per embedded test system preventive maintenance action.

It is clear from equation (4-45) that the costs are basically those related to the cost of hardware (embedded test systems, spare parts) and to the cost of maintenance manpower. Reductions in cost can only take place in these two general ares. Normally, an increase in the capability of an embedded test system to detect and isolate failures could be expected to increase the cost of developing and producing the test system hardware, while reducing the cost of auxiliary test equipment (less ETE required) and maintenance actions.

NOTE:
The value for MMH_{si} , average maintenance manhours as required for secondary isolation (to determine which of the $E(F_i)$ LRUs is the malfunctioning unit), can be calculated by various means, depending on the strategy used for troubleshooting/diagnosis.

1. If isolation is to be done by randomly testing or replacing the $E(F_i)$ LRUs, then:

$$MMH_{si} = (E(F_i)/2) * MMH_{sg} \qquad (4\text{-}46)$$

where:

MMH_{sg} is the average maintenance manhours required to determine that a given LRU is good or failed.

2. If a sequential troubleshooting guide is provided, the value of MMH_{si} is computed by taking into account the average manhours required to take each troubleshooting action, the troubleshooting sequence, and the relative probabilities of failure of each of the $E(F_i)$ LRUs. This probability may be determined by:

$$(\lambda_j)/\sum_{j=1}^{E(F_i)} \lambda_j \qquad (4\text{-}47)$$

where λ_j is the failure rate of LRU j belonging to $E(F_i)$.
When the embedded test system is designed to isolate to a unique LRU, MMH_{si} or MMH_s is 0.

4.2.3.5.2 Costs Associated with the Measures of Effectiveness
The imperfections of the test system (false alarm, failure to detect/isolate, etc.) can contribute substantially to the cost of maintenance manpower, associated with any test system design option. This section provides the mean cost of false removal, failure to diagnose, false alarm correction and expected number of removals measures at the O level. Using the testability tree diagram of Figure 4.2-1, all the decision branches are evaluated by their respective probabilities (probability of isolation, false alarm, false isolation, etc.). These probabilities can be estimated from the operational testing experiences of each test system. The outcome associated with each decision branch is also represented by a cost. These costs can be classified as follows:

- Cost of a test system of a particular design, and its implementation in mission time.

- Costs of maintenance (i.e. detection, isolation, repair, transportation, etc.) and those resulting from the uncertainty of the test system (i.e. failure, false alarm, false removal, failure to diagnose, etc.) in mission time.

We will begin with some additional notation.

- Notation.

Pr(SP) Probability that a spare part is available when needed at the O level.

Pr(ACM) Probability that a mission can be accomplished with any faulty LRU.

Pr(MT) Probability of terminating or aborting a mission as a result of CND.

C_{PS} Prime system/equipment cost.

C_{RR} Mean cost of removing and replacing an LRU from the system.

C_{ISOL} Mean cost of an LRU isolation by BIT after a potential failure is detected.

C_{TR} Mean cost of transporting the LRU between levels (this is the mean cost per LRU).

C_{MT} Mean cost of terminating the mission.

C_{GLRU} Mean cost of having a good LRU in the next level.

C_{MF} Mean cost associated with a failed mission (i.e., personnel and equipment)

a Costs associated with false removal.

The prime system is functioning properly, thus in the tree diagram, figure 4.2-1 we follow the path "prime system not faulty".
The related maintenance costs are:

- cost of removing and replacing the LRU
- cost of isolating the LRU
- cost of transporting the LRU to the next level
- cost of having a good LRU at the next level.

The other aspects to consider are spares and mission related costs. Using the cost diagram given in Figure 4.2-2, the costs associated with false removal C_{FRMV}, are:

$$C_{FRMV} = (FRMV)_O \{ [Pr(SP) + Pr(ACM) * (1- Pr(SP))] * [C_{ISOL} + C_{RR} + C_{TR} + C_{GLRU}] + (1- Pr(SP)) (1- Pr(ACM)) * [C_{MT} + C_{ISOL} + C_{RR} + C_{TR} + C_{GLRU}] \} \quad (4\text{-}48)$$

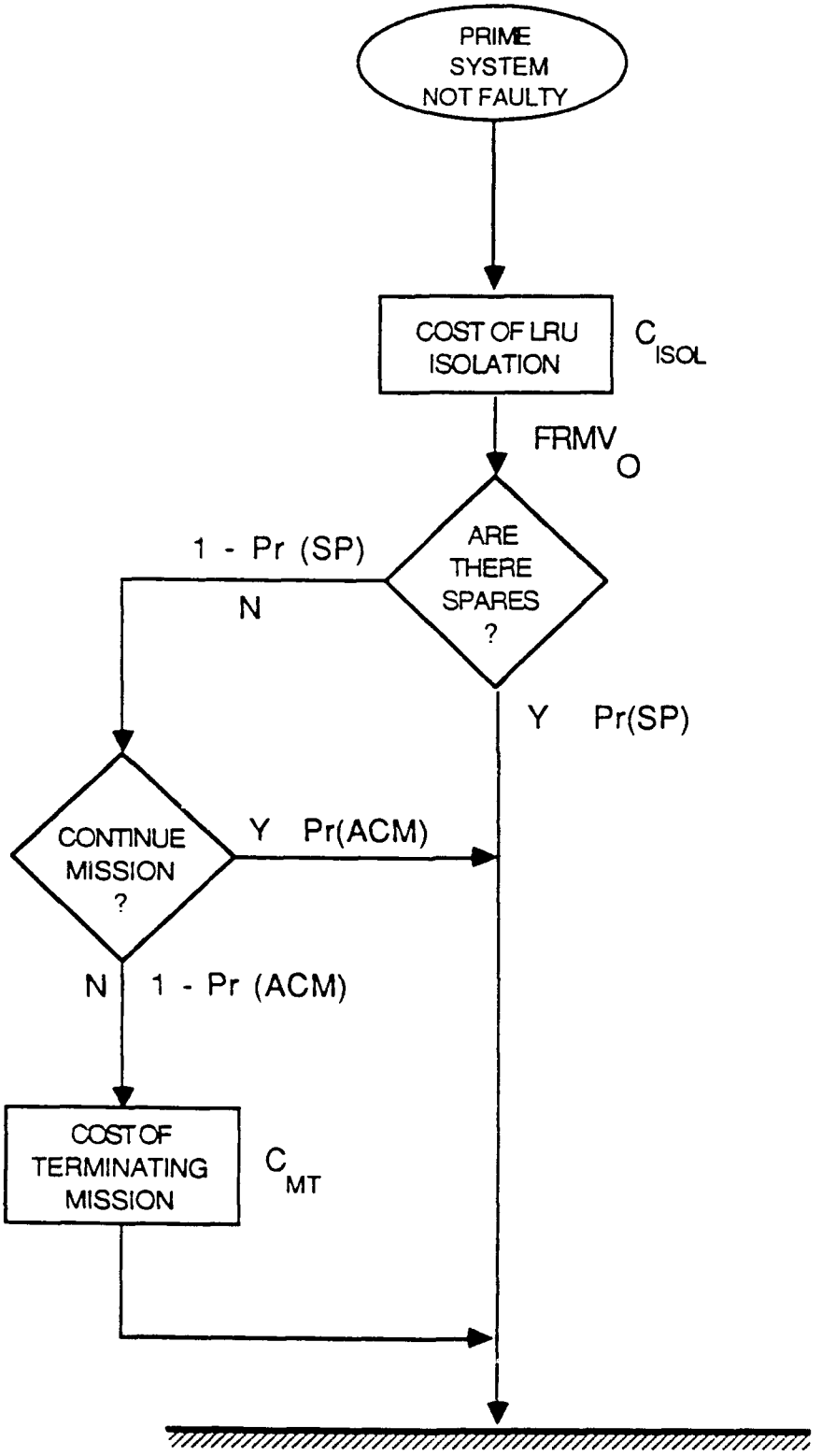


Figure 4.2-2 Cost Associated with False Removal (Due to False Alarms)

b Costs associated with FDG.

The "prime system is faulty" path is followed in the tree diagram of figure 4.2-1. The following cost diagram given in Figure 4.2-3 shows all the costs associated with failure to diagnose (FDG), at the O level and their respective probabilities. The maintenance costs are similar to the ones associated with false removals. Thus, following the same reasoning as for the derivation of FDG in section 4.2.2.3.2, the mean cost associated with FDG is given by:

C_{FDG} = (costs associated with failure to report)
*(Probability of failure to report)

+(costs associated with failure to isolate the LRU)
*(Probability of failure to isolate the LRU)

+(costs associated with incorrect LRU isolation)
*(Probability of incorrect LRU isolation)

Thus,

$$
\begin{aligned}
C_{FDG} = {} & Pr(F)_O\,[1- Pr(FD)_O]\,[1- Pr(ACM)]C_{MF} \\
& + Pr(F)_O\,Pr(FD)_O[1- Pr(FI_1)_O - Pr(FL)] \\
& * [\,C_{ISOL} + (1- Pr(ACM)\,C_{MF}] \\
& + Pr(F)_O\,Pr(FD)_O\,Pr(FL)_O * [(1- Pr(ACM))(C_{ISOL} + C_{MF}) \\
& + Pr(ACM) * (1- Pr(SP))\,(\,C_{MT} + C_{ISOL} + C_{RR} + C_{TR} + C_{GLRU}\,) \\
& + Pr(ACM)\,Pr(SP)\,(\,C_{ISOL} + C_{RR} + C_{TR} + C_{GLRU}\,)\,]
\end{aligned}
\qquad (4\text{-}49)
$$

Note that C_{FDG} is computed here using the isolation ambiguity group size 'i' equal to 1.

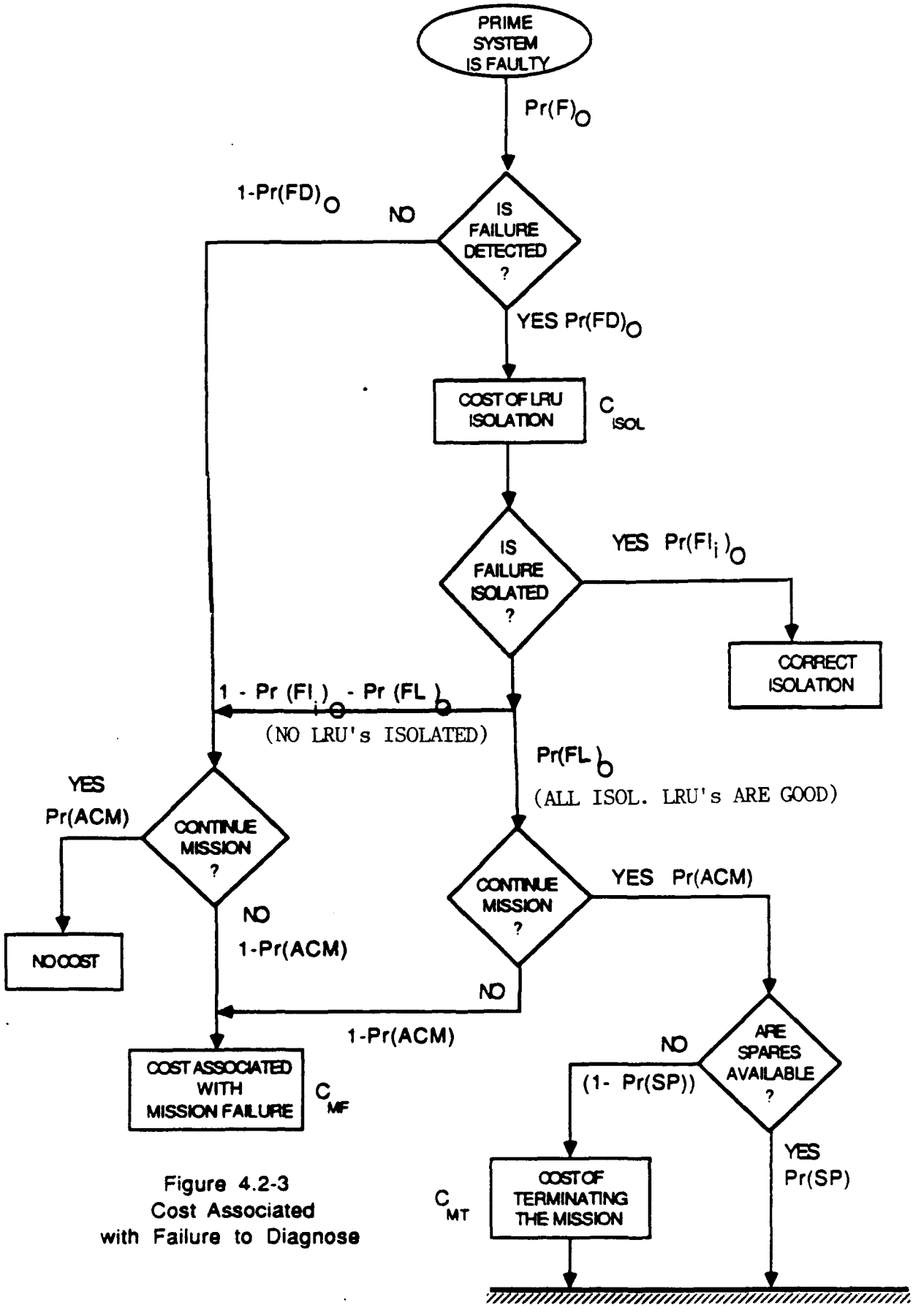


Figure 4.2-3
Cost Associated
with Failure to Diagnose

c Costs associated with FAC.

The cost diagram of Figure 4.2-4, associated with false alarm correction is derived from the tree diagram of Figure 4.2-1, by following the "Prime System is not faulty" path leading to the CND event. The costs associated with the correct action are the unnecessary isolation costs (maintenance), and the costs of terminating the mission due to the CND. Thus,

$$C_{FAC} = [1 - Pr(F)_O]\, Pr(FA)_O\, (FAC)_O\, [\, C_{MT}\, Pr(MT) + 1] * C_{ISOL} \qquad (4\text{-}50)$$

where

$$(FAC)_O = 1 - Pr(I \mid FA)_O .$$

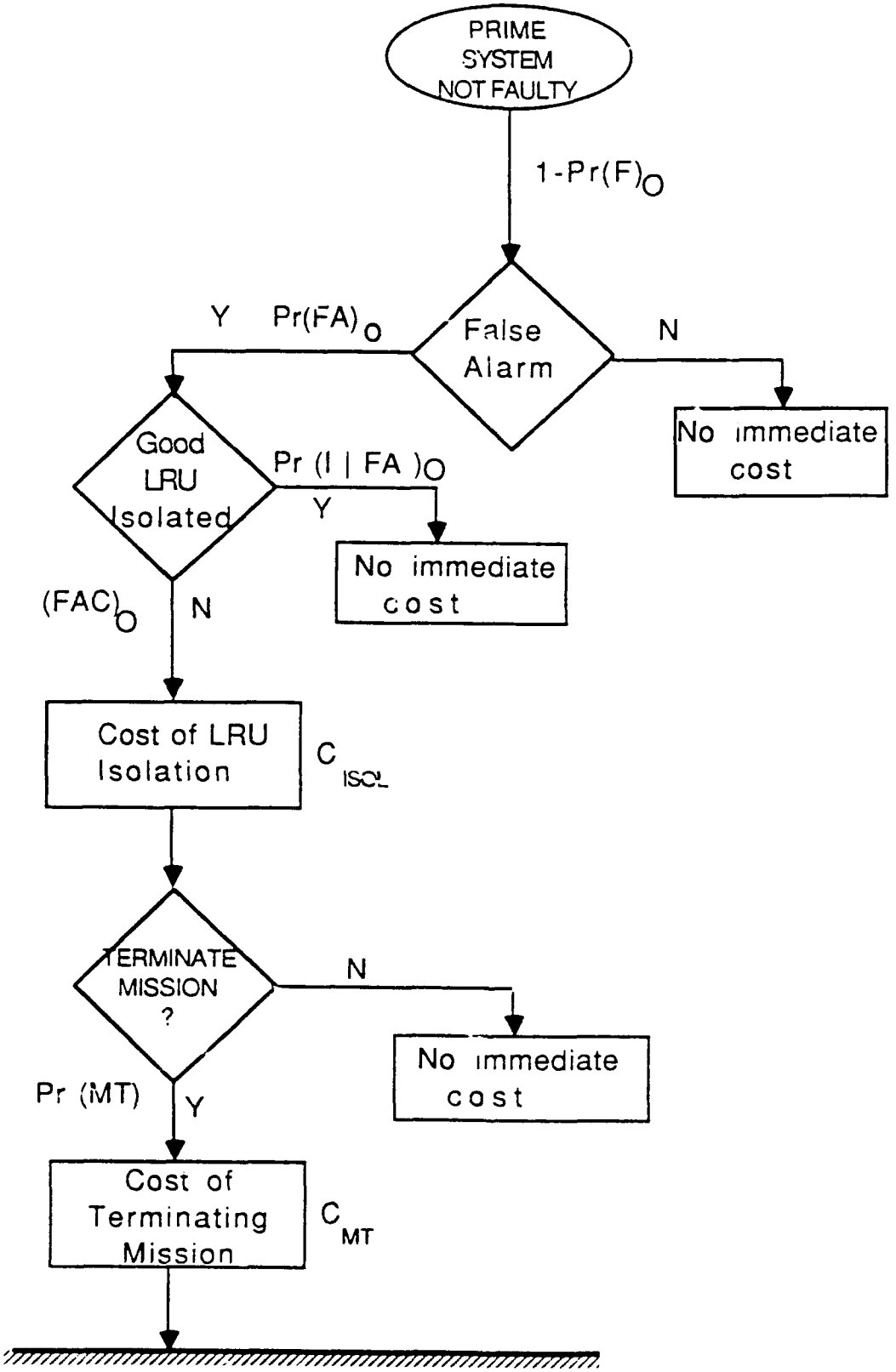


Figure 4.2-4 Cost associated with False Alarm Correction

d Costs associated with expected number of removals.

The ownership cost of weapon systems must include the effects of test system (BIT, ETE, etc.) requirements (FD, FI, erroneous isolations) on Operation and Support costs. Most LCC studies implicitly involve trading off certain performance requirements in order to balance Acquisition costs versus Operation & Support (O & S) costs. The Manpower and Spares costs are the two main cost drivers in operating and supporting military equipment. A major contributor to these cost drivers is the impact of diagnostic systems on manpower and spares costs. It is shown that with the value of E(RMV) that results from the diagnostic system requirements, it is possible to relate this measure of system effectiveness into support cost estimates for maintenance manpower and spares. It should be noted that the E(RMV) at the I and D level also contribute significantly to these costs. However, this study is restricted to the O level.

The approach used here assumes that at the O-level, line replaceable units (LRU's) are removed and then forwarded to the next level for further fault isolation. The repaired LRU's are then sent to organizational supply and the removed LRU's are sent to depot for further isolation to the piece part and repair.

- Costs of maintenance manpower CMMP.

These costs are based on the total repair time spent in returning failed systems to working order. A major portion of this time is spent in fault isolating the failed LRU, hence good FD/FI and a low false alarm rate will minimize the mean time to isolate (MTTI). Let us define,

SOH system operating hours

λ_S system failure rate (sum of the failure rates of all the components of the system)

TPMH average maintenance manhours required for test preparation at the O-level.

TMH — average maintenance manhours to perform a test at the O-level

E(RMV) — expected number of removals per failure at the O-level

RRMH — average maintenance manhours required to remove, replace and check at the O-level

LR — labor rate at the O-level, direct labor only.

The following equation is used to compute CMMP

$$CMMP = SOH\ \lambda_S\ [\ TPMH + TMH + E(RMV)\ RMMH\ ]\ LR \qquad (4\text{-}51)$$

- Costs of spares CSP.

A cost effective policy to achieve a desired availability for a system depends on the following factors for each replaceable unit:

- demand rate
- cost
- turnaround time
- repair time

The demand rates are spare parts consumption factors designating the average number of removals and replacements required per flight hour. As noted earlier, the lower the value of E(RMV), that is, the more effective the test system is, the less replaceable units will be required to support the system. The following equation gives the spares cost, which is simply the quantity of required spares (QSP) multiplied by the cost of the replaceable unit (C_u), i. e.,

$$CSP = QSP * C_u \qquad (4\text{-}52)$$

The number of spares have been determined in many ways. The most common method uses a "confidence level" (also called confidence factor or

probability of no stockout) as the determining criteria in spares calculations. This confidence level is the desired probability that all spare demands will be met for the item being considered within the item's turnaround time. Denote this probability by P. The turnaround time is measured from the time the item fails until the time it (or a replaceable unit) is returned and ready for use. In calculating P, it is generally assumed that the item's time between failure (demands) is expenentially distributed, i.e., its failure rate is constant. This assumption is usually valid for electronic components once they have gone through the early stages of their life. When the time between demands is exponential, it is known that the number of demands within the turnaround period follows a Poisson distribution. Thus, P is the probability that the number of demands is less than or equal to the number of spares, QSP. The parameter used in the Poisson distribution is the expected number of demands used during the turnaround time. This is often referred to as the product $n\lambda t$ and denoted by L. The formula for P is given by:

$$P = \sum_{k=0}^{QSP} \exp(-L)L^k/k! \qquad (4\text{-}53)$$

where
- P = Probability of meeting all spares demands within the turnaround time.
- QSP = Number of spares
- n = Quantity of items used for operating
- λ = Item failure rate in failures per unit time
- t = Average turnaround time

In our case, the parameter used in the Poisson distribution is given by:

$$L = SOH\ \lambda_S\ E(RMV) \qquad (4\text{-}54)$$

where all the parameters are defined as previously.

In order to determine the value of QSP, equation (4-53) may be solved iteratively, increasing QSP (i.e. , QSP = 0, QSP = 1, etc.,) until P or the

right half of the equation is greater than or equal to the specified probability or desired confidence level. In practice, the Poisson may be approximated by the Normal distribution to determine QSP as follows:

$$QSP = L + SL\,(L)^{1/2} \qquad (4\text{-}55)$$

where:

SL is the normal distribution value corresponding to the desired probability. This factor could be interpreted as ensuring that operational readiness requirements are met with a reasonable probability. The number of spares QSP is rounded up to the next integer value. For a probability of .90 or .95 that demand will not exceed spares, SL equals 1.282 and 1.645 respectively.

The formula for the spares cost is therefore,

$$CSP = \left\{SOH\,\lambda_S\,E(RMV) + SL\,(SOH\,\lambda_S\,E(RMV))^{1/2}\right\}^{*}\,C_U. \qquad (4\text{-}56)$$

4.2.3.5.3 Life cycle cost.

MIL-STD 1591 cost model has been shown to provide the overall cost of having embedded test systems FD/FI. However, the following are some inadequacies and discrepancies in the model:

(1) Cost benefits of having embedded test systems (e.g., reduced ETE) are not allowed.

(2) Costing different test system designs is not allowed (e.g., embedded test system fault is isolated to the SRU level versus fault detection only)

(3) It does not describe the characteristics and effects of embedded test systems (e.g., CNDs, RTOKs)

(4) In one part of the equation, it assumes that the embedded test systems are always available to detect system failures, while at the same time it computes the cost of repairing the test systems, hence a discrepancy.

In order to evaluate and assess the performance of alternative test systems (BIT, ETE, etc.), the costs associated with the inadequacies of the test systems, such as, FRMV, FDG, FAC should be integrated in the cost equation. This can be done by combining these costs into one figure representing the life-cycle cost of the test system.
The LCC of the test system is derived from the costs associated with false removals, failure to diagnose, false alarm correction and the probability of their occurrences (that is, the cost of using the test system per mission) as well as the prime system cost. The LCC is given by:

$$LCC = C_{PS} + \mu [C_{FRMV} + C_{FDG} + C_{FAC}] \qquad (4\text{-}57)$$

where

μ is the mean number of missions during the life cycle of the test system.

4.2.3.6 Maintenance Manpower Requirement.
The following model is provided for computing maintenance manhour requirements, a parameter which may be a constraint. The elements considered in this model may be described as follows:

- Total life cycle manhours spent in fixing faults detected by the test system.
- Total life cycle manhours spent in fixing faults not detected by the test system.
- Total life cycle manhours spent in fixing a failure in the test system itself.
- Total life cycle manhours spent in doing preventive maintenance on the test system.

Taking into account direct maintenance manhours only, total maintenance manhours is given by:

$$\begin{aligned} MMH_{TOT} &= (1 - P_F)\,[\,N_F\,\lambda_{PE}\,T\,Z\,(MMH_i + MMH_s)\,] \\ &+ P_F\,(N_F\,\lambda_{PE}\,T\,Z)\,[(MMH_{RP})\,] \\ &+ N_F\,\lambda_I\,T\,Z\,(C_{IFMP}) \\ &+ (\,N_F\,T\,Z\,/\,T_{PM}\,)\,(MMH_{PM}) \end{aligned} \qquad (4\text{-}58)$$

where all the variables are as defined in section 4.2.3.5.1

4.2.3.7 Overhead Burden Requirements.

These requirements represent the increase in costs associated with testability being incorporated into the system. These are the costs of the added fuel consumption due to the weight incurred by incorporating testability (e.g. hardware) into the system. This added weight will in turn affect the performance of the host system. These costs are included to ensure that the cases in which the weight of test systems may be too large are penalized accordingly. Thus, these costs do not apply at the detailed level of avionic design, but apply only to the system or subsystem level where the total weight of testabilty features (i. e., hardware) is too large. To estimate these costs one must have an estimate of the increase in aircraft weight due to the addition of the testability feature, the cost per weight per time flying, and the total time flown by all aircraft over the system's life cycle. These costs consist of the following:

- weight burden
- volume burden
- power burden
- cooling burden
- acquisition cost burden

All of these costs will be described in the following section on the top-down BIT prioritization.

Commentary on these relationships.

We have developed relationships between TFOMs, test systems (or resources) and system performance (R, P_{OR}, LCC, etc.) requirements.

A methodology has been developed to evaluate and assess the performance of the test systems or resources (BIT, ETE, etc.). The methodology combines the testability requirements for FD, FI and erroneous test system indications into measures of overall test system effectiveness. These measures are:

- False removal
- Failure to diagnose
- False alarm correction
- Expected number of removals per failure

These measures of effectiveness show the accuracy and precision of the diagnostic system and cover its inadequacies. These in turn, will enable the decisions to be made in order to evaluate and compare the performance of the mix of diagnostic resources based on their life cycle cost. This mean life cycle cost was predicted using the first three measures of effectiveness defined above, including the costs associated with the errors (false alarm, false isolation and failure to detect/isolate) of the test systems. In addition, the translation of the expected number of removals per failure, E(RMV), into Operation and Support (O & S) cost elements (manpower and spares cost) was also derived. Thus, tradeoffs between diagnostic systems capabilities and O & S out-lays can be made. Finally, in order to minimize the cost of acquiring diagnostic systems, and maximize effectiveness, these tradeoffs must be performed.

4.2.4 Top-Down BIT Prioritization.

The top-down approach to BIT prioritization has concentrated on interrelating design and mission parameters that involve BIT in order to establish a basis for an optimum application of BIT in the system design. The top-down approach to BIT prioritization reduces to a problem of optimizing or allocating the resource BIT used in the system design, throughout the levels of system indenture. Many of the relationships developed in previous section can be applied in the special case where BIT is the diagnostic system used to achieve the testability requirements. In the operational readiness equation, the effects of BIT on fault detection and fault isolation were shown to influence maintainability as given by mean time to repair (MTTR) , or mean time to corrective action (M_{ct}). Key parameters are BIT-influenced fault detection levels and BIT-influenced fault isolation levels as determined by failure rates and repair times. In effect, the top-down approach to BIT prioritization is a subset to the TAM problem, and hence it is treated within this integration phase.

The BIT allocation techniques employ criteria such as failure rates, mission criticality, MTTR, incremental weight burden, incremental life cycle costs, additional overhead burden requirements as defined in section 4.2.3.7, as well as any meaningful TFOM (e.g., BIT measure of effectiveness) . The pertinent aspect of the top-down approach is that it begins with mission and system requirements and applies these to the allocation either through the selection of an objective function or constraints relative to the BIT prioritization.

The approach taken in the BIT allocation is comprised of the following steps:

4.2.4.1 BIT Effectiveness vs System Parameters

4.2.4.2 BIT/BITE Relative Overhead Burdens

4.2.4.1 BIT Effectiveness vs System Parameters.

This section provides analytic procedures and mathematical tools which permit the system designer to specify the BIT requirements in precise, calculable and measurable engineering terms. Recall, that the primary objective of BIT is to correctly detect system failures and accurately

isolate the fault to a single LRU. The complex interactions between BIT performance parameters and system reliability, availability, maintainability and life cycle costs are also analyzed. This would permit BIT design tradeoffs to be performed meaningfully. The BIT performance parameters (Pr(FD), Pr(FI)), as described in section 2.0, and section 4.2.2 are subject to interpretation, difficult to measure and hard to achieve, in an operational environment. Thus, we need a BIT performance parameter that reflects the criteria stated above. Before selecting this parameter, let us note the following.

Maintenance-data systems provide the numbers for:

(1) remove and replace (RR) actions resulting from identified failures

(2) number of unnecessary removals (UNRMV) identified later on

(3) adjustments actions (REP).

Data provided from such maintenance-data systems are suspect when used. The reason is that there are many factors inherent in these systems which distort the reporting of these data. However, we can still define a measure of BIT effectiveness (or BIT inability)which satisfies the criterion of measurability using the information obtained from the maintenance-data systems. The measure of the inability of BIT (IN_{BIT}) to perform its task can be expressed as the ratio of the total number of maintenance actions, MA, (RR + UNRMV + REP) to the number of LRUs truly faulty (RR + REP), that is,

$$IN_{BIT} = (MA)/(MA - UNRMV) \qquad (4\text{-}59)$$

The aim therefore is to minimize this measure (IN_{BIT}), that is, every true failure should result in <u>one</u> maintenance action. In order to accomplish this, we note that, from equation (4-59) the number of unnecessary removals should be minimized (UNRMV = 0). However, the UNRMV is due to the following:

- the inability of the diagnostic system (BIT) to correctly detect and isolate faults (*i.e., failure to diagnose, FDG*)

- False alarms (FA)

- Isolation to an ambiguity group (instead to one single unit)

Therefore, to minimize the number of unnecessary removals, we must minimize every measure associated with each of the factors defined above.

Thus, in addition to the measures of effectiveness of test systems given in section 4.2.2.3, which can be used when BIT is the test system, we have found another measure of BIT performance, namely, IN_{BIT}. The inability of BIT to perform its task as designated, satisfies the criterion of measurability during the overall system design process. The relationship between this measure and the prime system performance parameters (R, A, and LCC) follows.

4.2.4.1.1 Reliability vs BIT:
The addition of a BIT feature will increase the system complexity and hence the failure rate, to the degree that components and/or software are added to accomplish this testability function. Thus, the impact of the testability on system reliability expressed as the change in failure rate may be given by:

$$FOM_{BIT} = \lambda_S / (\lambda_S + \lambda_{BIT}) \qquad (4\text{-}60)$$

where:

λ_{BIT} = failure rate due to the BIT feature, that is the circuitry, and/or software added.

The FOM_{BIT} factor will also affect the O-level mean time between failure ($MTBF_O$) based on true maintenance actions, that is, the $MTBF_O$ becomes $FOM_{BIT} * MTBF_O$. It should be pointed out that the term FOM_{BIT} must be included in the prime system's reliability calculation.

4.2.4.1.2 Maintainability vs BIT.

The inability of BIT (IN_{BIT}), reflected in the number of unnecessary removals, will drive the maintainability of the system as determined by MTTR, or M_{ct}, as a function of IN_{BIT}, depending on the support, maintenance and logistics concepts. Assuming a linear function which will provide sufficient accuracy, the M_{ct} is modified as follows:

$$M'_{ct} = M_{ct} * IN_{BIT} \quad (4\text{-}61)$$

4.2.4.1.3 Availability vs BIT.

Availability is mathematically defined as:

$$A = MTBF_O / (MTBF_O + M'_{ct}) \quad (4\text{-}62)$$

$$A = (1/(1 + (M'_{ct} / MTBF_O)) \quad (4\text{-}63)$$

where $MTBF_O$ and M'_{ct} are as previously defined. As stated previously in the reliability and maintainability calculations, the impact of BIT will modify the availability equation (4-63). Substitution of (4-60) and (4-61) in (4-63) yields:

$$A = 1/[\, 1 + (\, M_{ct} * IN_{BIT}) / (MTBF_O * FOM_{BIT})] \quad (4\text{-}64)$$

Note: Equation (4-64) shows that both M_{ct} and $MTBF_O$ are sensitive parameters in the computation of availability. The impact of the BIT system performance (IN_{BIT}) on the availability characteristics of the system is also shown. The ratio (IN_{BIT} / FOM_{BIT}) suggests that there may be some regions where the improvement in BIT performance (i.e., minimizing IN_{BIT}) is offset by the change in BIT reliability (FOM_{BIT}).

4.2.4.1.4 LCC vs BIT.

The reasons for using a cost model are to provide uniform criteria and a consistent cost accounting structure for LCC evaluation and trade-off studies. The MIL-STD-1591 model, as defined in section 4.2.3.5.1, is a model which provides criteria for conducting trade-studies and the

analysis/synthesis of aircraft BIT using an LCC model. The standard cost elements in the pre-programmed equations are conveniently provided to the system designer. The designer, in turn is assured that all relevant cost elements have been included, so that the selection of alternate BIT concepts and design features through LCC trade-offs is meaningful. Therefore, using criteria similar to the MIL-STD-1591, a model for BIT LCC trade-off is developed in this section.

a. Model Cost Equations.
The model evaluates the elements of LCC which are sensitive to the BIT features. These features may be defined as:

- BIT vs ETE
- BIT hardware vs BIT software
- Centralized BIT vs decentralized BIT

All the other elements of the prime system's LCC which are insensitive to the BIT features are excluded. In addition, the model cost equations are simplified to minimize the requirements for the extensive input data which usually characterizes LCC models. Thus, only the relevant cost elements that are useful in analyzing the cost impact of BIT features are considered in this model. The model uses LRU level performance and design parameters. The LCC elements for each LRU are automatically summed in the model to evaluate subsystem trade-offs. In general, implementation of a given BIT feature or features will affect the classic elements of LCC of the prime system, as defined in section 4.2.3.5.1. These elements are:

- Research and Development
- Acquisition Costs (which includes Production and Initial support cost)
- Operation and Support (O & S) Costs

Thus, the incremental change in life cycle cost is the sum of the incremental changes in these cost categories. In the equations that follow,

the terms are defined as elements of incremental change in these categories. Therefore, the total life cycle cost incremental change is given by:

$$LCC_T = C_R + C_A + C_{OS} \quad (4\text{-}65)$$

where:

LCC_T = Total life cycle cost incremental change

C_R = Incremental cost of Research and Development

C_A = Incremental Acquisition costs

C_{OS} = Incremental Operation and Support costs

A discussion of each of the cost factors identified above is presented in the following paragraphs.

b. Incremental Cost Of Research and Development.
This cost reflects the implementation of new test techniques that require research and development. In most cases, this term will be null. However, there must be cases where a specific testability feature might require the development of a new type of sensor to implement a test point, in such cases this cost will be the estimated cost of these efforts. Estimates must include costs to develop both new advanced state-of-the-art hardware as well as any required software.

c Incremental Acquisition Costs.
This category includes all incremental production costs (C_P) and incremental initial support costs (C_{IS}), and is given by:

$$C_A = C_P + C_{IS} \quad (4\text{-}66)$$

d Incremental Production Costs.
The incremental production costs include both recurring and nonrecurring

costs of BIT hardware and software. These costs include the following factors:

- incremental cost of design
- incremental cost of manufacturing

• Incremental Cost Of Design. (C_D)

The incremental design cost C_D of a BIT feature is an increment of cost added to the cost of the subsystem (or LRU) because of the implementation of the BIT feature. In avionics, BIT is used at the O level in order to speed up fault isolation and modular replacement which in turn will help achieve higher availability. BIT design engineering has become quite sophisticated and normally uses the operational hardware to test itself, as opposed to simply using remotely controlled test equipment. The result has been that the amount of BIT-dedicated hardware required has not increased in proportion to the percentage of Fault Detection (FD) achieved. In deriving the design cost, it will be easy to achieve a certain percentage of FD, say 80%, but to increase that to 90% or more the cost will be great. In other words, the function of a percentage of FD vs the design cost will be an exponential curve. A study made by Westinghouse [24] confirms also this fact. Thus, the incremental design cost is given by:

$$C_D = \exp(a * FD) * C_{LRU} \qquad (4\text{-}67)$$

where:

a = coefficient obtained from experience or varied as a variable.

C_{LRU} = cost of a single LRU.

Experience indicates that a high percentage of FD is achievable if very close interaction with hardware/software design is enforced. Thus, the coefficient a is a function of state-of-the-art in terms of design engineering, and a value between 0 and 1 will be suitable. In fact a value of a close to 1 will be reasonable due to the sophistication of BIT design engineering.

• Incremental Cost of Manufacturing.

The manufacturing cost, C_M, is the cost of a projected BIT feature, and therefore it becomes an increment of cost to the system. This cost is computed as follows:

$$C_M = N (PC + LC) + N (PC + LC) AC$$

$$C_M = N (PC + LC) (1 + AC) \qquad (4\text{-}68)$$

where:

N = number of production subsystems (LRU's) with BIT feature incorporated, that is,

N = no. of subsystems * no. of LRUs per subsystem.

PC = Purchase cost of parts and material for the BIT feature per LRU.

LC = Labor cost necessary to fabricate the BIT feature per LRU.

AC = Administrative cost of adding the BIT feature to the production process per LRU, expressed as a fraction of production costs.

e. Incremental Initial Support Costs.

These costs are the sum of the incremental change in the cost of test equipment and test software (C_{TESW}) as the result of that BIT feature, plus the incremental change in the cost of initial spares (C_{ISP}) resulting from that BIT feature. Thus, the equation is:

$$C_{IS} = C_{TESW} + C_{ISP} \qquad (4\text{-}69)$$

• Cost of test equipment and software.

A given BIT feature implementation may influence the cost of test equipment and software required to support the system. This cost is based on an estimate of the impact (if any) that the testability capability (BIT) may have on a set of test equipment (TE) and software, and is given by:

$$C_{TESW} = CSW + M * LOC (C_{TEBIT} - C_{TENOBIT}) \quad (4\text{-}70)$$

where:

CSW = cost change in software design due to the BIT feature.

M = number of test equipment sets per location.

LOC = number of Locations to be outfitted.

C_{TEBIT} = cost of a single set of test equipment to support the prime system with the BIT feature incorporated.

$C_{TENOBIT}$ = cost of a single set of test equipment to support the prime system with the BIT feature not incorporated.

The software cost is driven by "testability" of a given design. Testability means or capability (e. g., BIT, ETE) has the effect of improving this testability by adding, test points, control inputs, sensors, etc. Therefore, to estimate this change in cost we would consider the influence the hardware implementation of this BIT feature will have on the prime system. To estimate the cost of test equipments, we have to take into account the complexity of the test equipments circuitry, and their intrinsic testab'lity, since the addition of BIT circuitry requires means to test that same circuitry. It should be noted that a given BIT implementation will result in a cost reduction of test equipment.

• Incremental Cost of Initial Spares.

The cost of initial spares required to support a number of systems may

change as a result of a given BIT feature. This testability feature affects the number of subsystem removals which in turn impacts the number of spares. Therefore, the cost of spares must account for both the inability of BIT (IN_{BIT}) , and the change in subsystem reliability due to the BIT feature. The initial cost of spares is therefore given by:

$$C_{ISP} = LOC(CSP_{BIT} - CSP_{NBIT}) \qquad (4\text{-}71)$$

where:

CSP_{BIT} , CSP_{NBIT} = cost of spares per location with and without testability.

Using equation (4-52), we get:

$$CSP_{BIT} = QSP_{BIT} * C_u \qquad (4\text{-}72)$$

and

$$CSP_{NBIT} = QSP_{NBIT} * C_u \qquad (4\text{-}73)$$

where:

QSP_{BIT} , QSP_{NBIT} = number of spares per location with and without testability feature incorporated.

C_u = cost of the LRU

In the above equation, every quantity is an input data except for the number of spares (QSP). This is computed from equations (4-53), and (4-54) as follows:

$$P = \sum_{k=0}^{QSP} \exp(- L) L^k/k!$$

$$L = SOH\ \lambda_{LRU}\ E(RMV) \qquad (4\text{-}74)$$

where:

L = number of removals

E(RMV) = expected number of removals per failure.

The other parameters are defined as previously.

In the above equation for L, we see the relationships between the reliability, and the BIT effectiveness, and how the number of spares may vary as a function of LRU removal. Note that, in the reliability expression, λ_{LRU}, we have to factor out the change in failure rate due to BIT, that is, the expression given by equation (4-60):

$$FOM_{BIT} = \lambda_{LRU} / (\lambda_{LRU} + \lambda_{BIT}) \qquad (4\text{-}75)$$

f Incremental Operation and Support Costs.

These are all costs of system ownership and operation. They encompass total personnel and material costs necessary to operate and maintain the prime system over its life cycle. However, only base maintenance cost is significantly impacted by BIT. It is the product of the costs per maintenance manhours and the number of maintenance manhours used. Implementation of BIT features will change the required maintenance effort by affecting the total number of maintenance hours that a system will require during its operational life. The change in total maintenance hours is equated to a change in O & S costs and is given by:

$$C_{OS} = MMH_O * C_L \qquad (4\text{-}76)$$

where:

MMH_O = change in O level maintenance manhours due to the BIT features incorporated, i. e., change in life cycle direct maintenance manhours.

C_L = total cost of a direct maintenance manhour.

Testability via BIT will in most cases increase the failure rate of the system due to the addition of hardware/software. At the same time, the Testability capability (e. g., BIT) will reduce the amount of effort needed to bring back the system into operation, by improving the quality of maintenance. Thus, the change in O level maintenance is given by:

$$MMH_O = L * (\lambda_{BIT} M'_{ctBIT} - \lambda_{NBIT} M'_{ctNBIT}) \qquad (4\text{-}77)$$

where:

L = total flight hours over the life cycle which is, the product of the number of systems to be built, the number of years of operational life of these systems, and the yearly operating hours of a single system.

λ_{BIT} , λ_{NBIT} = are failure rates with and without the BIT feature,

M'_{tBIT} , M'_{ctNBIT} = O level mean corrective times for the LRU with and without the testability feature, given by equation (4-61), that is, $M'_{ct} \approx M_{ct} * IN_{BIT}$

4.2.4.2 BIT/BITE Relative Overhead Burdens.

The following section describes the procedure used in the MATE GUIDE dated 1 April 1985 (G3V3P2 APPENDIX E: Testability Overhead Estimation, and G3V3P2S7) for the computation of the hardware relative overhead burdens of BIT/BITE testability features for various levels of probability of fault isolation for any particular avionics equipment subsystem.

Note: In what follows, all the figures related to the MATE Guide are referenced using the same numbering scheme as in the MATE Guide (i.e. Fig. E-X) and are to be found in Appendix A. To keep the numbering logical and consecutive, the tables are referenced differently (i.e., T-x instead of E-x). The first six of the tables are also taken from the MATE Guide and can be found in Appendix A.

The approach taken in the derivation of these burdens is comprised of the following steps:

4.2.4.2.1 Definitions and Assumptions

4.2.4.2.2 Computation Procedure

4.2.4.2.3 Application

4.2.4.2.1 Definitions and Assumptions.
This step provides the definitions of terms used followed by the assumptions made in the computation of the hardware burdens.

• Definitions.

a. Relative Overhead Burden:
"Relative Overhead (burden) of BIT/BITE is a measure of the technical assets built into an avionics functional unit to provide testability features " (MATE GUIDE G3V3P2S4). These features (hardware/software) are those design characteristics of a functional unit which enhance FD/FI. The implementation of these features, however, must be traded-off against the increased development costs, along with the changes to the avionics environment (weight, volume, power and cooling.)

b. Hardware Burden:
This is the incremental increase in the prime system functional configuration, as a proportion of the unburdened prime equipment attribute, to achieve the fault isolation performance (probability of isolation) required.

c. Test Difficulty Factor:
The Test Difficulty Factor (TDF) is defined as (or based on) the relative time (complexity) to run end-to-end test for fault isolation.

d. LRU Modularity Factor:
The LRU Modularity Factor (LRUF) reflects the burden of BIT in performing the functional end-to-end test which constitutes fault detection.

• Assumptions.

The hardware burdens computed are based on the following assumptions and conditions:

a. The hardware burden is, in fact, hardware and firmware burden. It does not include any true software considerations except for subsystems which historically include processors.

b. The analog-to-digital (A/D mix) relationship used to determine the hardware burden is based upon the estimated manufacturing cost ratio between these functional aspects of the avionics subsystem.

c. The subsystem lot size is 500 units, which represents the recurring production costs.

d. Actual BIT cost in dollars is derived by inputting the hardware burden data into a life cycle cost model.

e. The hardware burden to achieve various probability of isolation levels has been developed by reviewing historic data on existing equipments in the Air Force inventory as well as assessing (then) current (pre-1985) design practices. (Note: This data requires periodic update to be kept valid.)

f. The Relative Hardware Overhead for BIT/BITE can be used to calculate the additional overhead burdens, namely, relative weight, volume, power and cooling burdens.

g. Probability of isolation used here is the absolute probability of fault isolation. This is defined as the product of the probability of detection and the conditional probability of isolation given that detection has occurred . The equation is:

$$Pr(I) = PD * PFI \qquad (4-78)$$

It is expressed as a decimal ratio to the equipment base. It includes both the hardware implementation and software implementation as expressed in firmware. The probability of isolation Pr(I) at the O level relates to the inherent testability of the avionics subsystem.

h. Probability of isolation ratios assume that all component failures and faults that contribute to a performance failure condition of the unit under test are included in the failure rate data base (i.e., "non-detects" are included in the data base to which the isolation ratio is addressed)

i. In determining the level of isolation, the impact of false alarm, Pr(FA), or probability of false detection on testability feature burdens has to be studied.

4.2.4.2.2 Computation Procedure.
This step is comprised of the following:

A. Total Hardware Burden (CBF) Vs Probability of Isolation Pr(I)

B. CBF Vs System Characteristics

C. System Characteristics Vs Probability of Isolation

A. Total Hardware Burden (CBF) Vs Probability of Isolation (Pr(I)).
The procedure described here sequentially calculates the hardware burden of testability features for each level in the proposed maintenance concept based on procedures provided in Guide G3V3P2, Appendix E. The process of computing the hardware burden of BIT/BITE testability features for various levels of isolation in avionics equipments utilizes a five-step procedure presented below. Figure E-1 in Appendix A depicts a flow diagram of this procedure.

step 1: Initial set-up of worksheet
The Test burden worksheet provided in Figure E-2 is initially set up with data specific to the avionics equipment as obtained from Tables T-1 and T-2. Table T-1 provides Specific Testability Requirements for generic avionics categories. The breakdown is given in terms of:

- Generic Avionics Category
- Avionic Equipment Group
- A/D Mix Factor
- Test Difficulty Factor
- Historic Number of LRUs.

Table T-2 LRU MODULARITY FACTORS FOR FAULT ISOLATION provides a factor for use in the computation of the hardware burden.

step 2: Derivation of O (Flight Line) level Hardware Burden of BIT (FLHB)
The hardware burden data (or uncompensated burden factor UCBF) is given in graphical terms for different levels of maintenance as a function of isolation levels and probability as well as analog-to-digital ratio of the equipment makeup. Using the two-level or three-level set of curves in Figures E-4 through E-9, along with the probability of isolation level desired, we select the UCBF of BIT from the respective curves for fault isolation to one, two or three LRUs/SRUs. Detail data points from these curves are given in Tables T-3, T-4, T-5, T-6. These tables list the data for both the LRUs and SRUs at the O level, SRUs at the I level, and the component groups of seven piece parts at the D level. Since this study is restricted to the two level of repair maintenance concept (O and D), we will use the the two-level "O" curves in Figures E-7, E-8 and E-9. As may be readily seen from the above mentioned curves the data is not a linear function of probability of isolation. These figures are also useful since they permit calculation for isolation percentages not listed in the tables T-3, T-4, T-5 and T-6. Thus, the Flight Line Hardware Burden, FLHB, is found by multiplying the UCBF by the TDF (given in Table T-1) and by the LRUF (given in Table T-2). This factor is expressed as:

$$FLHB = UCBF * TDF * LRUF \qquad (4\text{-}79)$$

step 3: Derivation of I level Hardware Burden
Since the two level of repair maintenance concept has been selected, the I level Hardware Burden does not apply here.

step 4: Derivation of the D level Hardware Burden (DLHB)
The DLHB is derived from the curves in Figure E-10 (MATE Guide Figure E-13) as a function of probability of isolation.

step 5: Derivation of the Total Hardware Burden.
This factor is calculated by summing the separate hardware burdens to arrive at the total hardware burden or compensated burden factor (CBF) for BIT/BITE. This is defined as, the cost relative to or the increase in cost over the existing hardware cost without BIT. It can be expressed as:

$$CBF = FLHB + DLHB \qquad (4\text{-}80)$$

NOTE: Using the LRU Modularity Factor definition and Table T-2, we can determine the CBF for fault detection to any subsystem. This is computed as follows: Select the appropriate UCBF of BIT/BITE from Figure E-4 with the A/D mix factor, multiply by the TDF for the subsystem and then by 0.2 (0.2 is the minimum hardware burden of BIT required to perform the functional end-to-end test which constitutes fault detection), the LRU Modularity Factor, that is,

$$CBF \text{ (for fault detection)} = UCBF * TDF * 0.2 \qquad (4\text{-}81)$$

B. CBF Vs System Characteristics.
This step consists in converting the hardware burden (CBF) data to the system characteristics of weight, volume, power and cooling. Techniques for relating the CBF to these requirements are derived from MATE Guide G3V3P2S7.

1. Weight Vs CBF.
The relationship between the CBF and weight is non-linear, as shown in Figure E-11 (Guide G3V3P2S7, Figure 7-2), and based on historical data. Detail data points from this curve are itemized in Table T-7. This relationship may be used directly or linearized without producing significant additional error.

WEIGHT	CBF
.05	.0364
.10	.0818
.15	.1190
.20	.1548
.25	.1905
.30	.2238
.35	.2500
.40	.2740
.45	.2900
.50	.3145
.55	.3387
.60	.3547

TABLE T-7 COMPENSATED BURDEN VS WEIGHT

Fitting the set of these data points to a straight line model we get Figure 4.2-5.

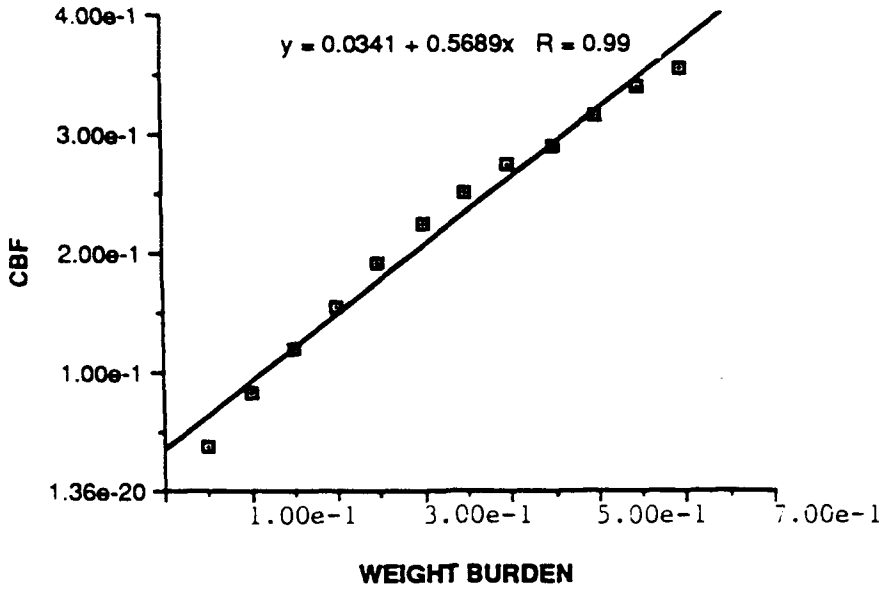


FIGURE 4.2-5 CBF VS WEIGHT BURDEN

Thus, the relationship between the weight burden and CBF is obtained by inverting the above equation yielding:

$$W = 1.7578 * CBF - .0599 \qquad (4\text{-}82)$$

2. Volume Vs CBF.

The volume burden is assumed to be directly proportional to the weight overhead for the BIT/BITE. Thus, data points from Table T-7 can be used where Volume is substituted for weight .

Thus, the equation relating the volume burden to the CBF is:

$$V = 1.7578 * CBF - .0599 \qquad (4\text{-}83)$$

3. Power Vs CBF.

Power (P) burden is calculated as the product of the power overhead factor, F_p, and the increase in hardware of BIT/BITE, that is,

$$P = F_p * \text{BIT/BITE overhead} \qquad (4\text{-}84)$$

The power overhead factor, F_p , is provided in Figure E-12 (MATE Guide G3V3P2S7, Figure 7-3). This curve shows a linear relationship between F_p and the A/D mix ratio (obtained from Table T-1) used to implement the BIT/BITE circuitry.
Thus, using E-12 this relationship can be expressed as:

$$F_p = (.55/60)\ (100 - (A/D)) + .05$$

$$F_p = -.00917 * (A/D) + .967 \qquad (4\text{-}85)$$

Substituting equations (4-82) or (4-83) into (4-84) we have:

$$P = F_p * (1.7578 * CBF - .0599) \qquad (4\text{-}86)$$

where F_p is given by equation (4-85).

4. Cooling Vs CBF.

The cooling ovehead factor (C) for BIT/BITE is obtained as the same value as for power burden determined above.

C. System Characteristics Vs Probability of Isolation.

This section summarizes the results from both part A and part B to determine the relationship between the probability of isolation (Pr(I)), and the system requirements such as, weight, volume, power and cooling.

From part A, we conclude that if we linearly fit the data points relating the CBF to the Pr(I) we get the relationship:

$$CBF = a * Pr(I) + b \qquad (4\text{-}87)$$

where a and b are constants which depend on the TDF, LRUF, A/D ratio and the particular subsystem.

In part B we saw that, if we denote the system requirements (W, V, P, C) by S, then the linear relationship between S and the CBF is given by:

$$S = c * CBF + d \qquad (4\text{-}88)$$

where:

$c = 1.7578$ (for W and V) or $F_p * 1.7578$ (for P and C)

$d = -.0599$ (for W and V) or $F_p * (-.0599)$ (for P and C).

Therefore, substituting equation (4-87) into (4.88) we get:

$$S = e * Pr(I) + f \qquad (4\text{-}89)$$

where e and f are constants to be determined in each case.

4.2.4.2.3 Application.
We show how to apply the computation procedure outlined in the previous paragraphs, in the special case of a Doppler Radar, for the various probability of isolation, Pr(I) (.88, .90, .95, .98). This will enable us to determine the constants a, b, c, d, e and f.
We will begin by computing the CBF.

- Computation of the CBF.

Since this sudy is restricted to the O level only, the CBF is the Flight line hardware burden (FLHB). Thus, equation (4-80) yields:

$$CBF = FLHB \qquad (4\text{-}90)$$

In order to compute the FLHB, we will show a step-by-step generation of a single overhead burden which, in conjunction with other data, may be iteratively used over a range of single inputs to form a table to generate the relationships between S and Pr(I).
Thus, we will try to expand one step in the procedure namely, the calculation of a single overhead burden for a single Ambiguity Group Size (AGS). (e.g., line 5 in Figure E-2 for an AGS of 1 SRU). In what follows, we will give the generic parts in each phase and apply to the example mentioned above.

This step is comprised of the following:

Phase I. Input Data.

The input data consists of the following:

	Example
1. Avionic Subsystem	Doppler Radar
2. Number of LRUs in the subsystem	3
3. Ambiguity Group Size (AGS)	1
4. Probability Of Isolation (Pr(I))	.88 (.90, .95, .98)
5. Maintenance Concept	O level

Phase II. Determination of the factors in FLHB.

This data is secured from the tables T-1 and T-2 found in Appendix A.

1. Table T-1 gives:

a. A/D ratio	30/70 (%)
b. Test Difficulty Factor (TDF)	1

2. Table T-2 gives:

a. LRUF for Isolation To

1 LRUs out of

3 LRUs = .67

3. Tables T-3 through T-6 give:

a. UCBF at the Flight Line for

a testability level of __.88__ and an

A/D ratio of __30/70%__ = .244

Phase III. Computation of CBF (FLHB).

Equation (4-79) gives:

CBF = UCBF * TDF * LRUF

Thus,

1. UCBF = .244

2. TDF = 1

3. LRUF = .67

CBF = .163

Therefore, if we apply this procedure to the various Pr(I), we get Table T-8:

Pr(I)	CBF
.88	.163
.90	.164
.95	.191
.98	.285

TABLE T-8 COMPENSATED BURDEN VS Pr(I)

Fitting these data points using a linear regression analysis, we get Figure 4.2-6:

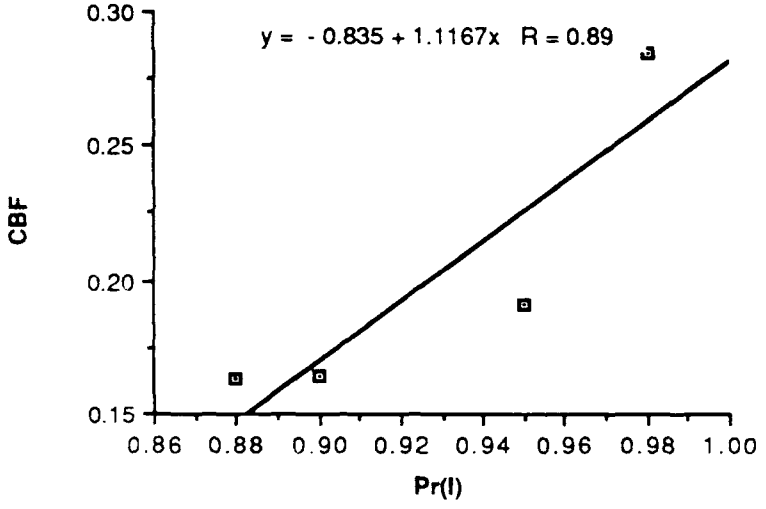


Figure 4.2-6 Least Squares Fit To CBF Vs Pr(I) For One LRU (DOPPLER RADAR)

NOTE: This graph shows an exponential relationship, however as a first step to using the TAM, we linearized the relationship between CBF and Pr(I).
Thus, the equation is:

$$CBF = 1.1167 * Pr(I) - 0.835 \qquad (4\text{-}91,$$

Comparing (4-91) to (4-87) we find:

$$a = 1.1167, \quad b = -0.835$$

Substituting equation (4-91) into (4-82) yields the following equation for the weight in terms of Pr(I) for the Doppler Radar:

$$W = 1.7578\ (1.1167 * Pr(I) - .835) - .0599$$

$$W = 1.9629 * Pr(I) - 1.5277 \qquad (4\text{-}92)$$

• Summary.

If the same analysis is made for all subsystems, the resulting equations are summed together and a curve showing the total weight burden for the total subsystem can be plotted. This equation relating the weight of the total subsystem in terms of the probability of isolation (TFOM) can be used as a constraint in the testability allocation problem.

4.2.5 Selection of Objective and Constraint functions.
The selection of "cost" functions depends on the levels of indenture (system, subsystem, LRU, etc) and the maintenance level (O and D).

At the system level, Design For Testability (DFT) is to be achieved by minimizing false flight line removals and retest OK (RTOK) or cannot duplicate (CND) events at the organizational maintenance level (O level). For this, we may require functional modular partitioning of the avionics system and the test equipment. In addition, it is imperative that the system design be such as to maintain low false alarm rates. High false alarm rates obscure the true faults detected and the true fault isolation indications and contribute to high RTOK and erosion of pilot and flight-line maintenance crew confidence in the BIT system.

At the subsystem level, the criteria include and extend the weapon system level criteria. It requires functional modularity within the subsystem or LRU and minimization of false alarms. It requires fault detection and isolation levels meeting weapon system operational requirements commensurate with acceptable overhead burdens and penalties in terms of weight, volume, power, cooling, reliability, test time, cost etc. For avionics subsystems, these requirements are addressed to the O level, i. e., pre-flight, in-flight, post-flight, scheduled and other flight-line maintenance test activities.

At the LRU level, testability is provided such that tests are repeatable from the O to the D level. Implementation of this concept will minimize "Retest OK" (RTOK) in the Depot.

The section organization and corresponding approach in the selection of "cost" functions is comprised of the following:

4.2.5.1 Problem Formulation

4.2.5.2 Choice of Objective and Constraint Functions

4.2.5.3 Application: Example Problems

4.2.5.4 Summary

4.2.5.1 Problem Formulation

The Testability allocation problem as stated earlier is viewed in two equivalent ways:

• Allocate the TFOMs (FD, FA, T_D, FI_P, FP, T_I) and/or any new developed TFOM (as defined in section 4.2.2) cost effectively across the levels of indenture to satisfy system requirements;

• Determine the optimal allocation of test resources (BIT/BITE, ETE, etc.) thereby the components TFOMs will then be determined optimally.

The testability allocation could be viewed as a bounded resource allocation problem, and is solved using the Augmented Lagrangian method. The fundamental characteristic associated with this method is the n-dimensional unconstrained minimization of a differential function that involves Lagrange multiplier estimates and a penalty term.
The testability problem has a separable structure that is, the objective and constraint functions depend on one TFOM at a time.

This study is restricted to the O level, thus, the allocation problem is set-up as follows:

$$\textbf{MIN} \sum_{i=1}^{n} f(i)\, X(i) \qquad (4\text{-}93)$$

subject to:

$$\sum_{i=1}^{n} g(ij)\, X(i) \leq C(j)$$

$$j = 1,\ldots\ldots,m$$

$$0 \leq X_i \leq 1$$

where:

f(i) & g (ij)	:	additive and separable objective and constraints functions (no TFOM cross product)
X(i)	:	amount of testability or coverage (TFOM) to be allocated, which is bounded
n	:	number of units (subsystems, LRUs) per level
m	:	number of constraints
C(j)	:	amount of jth resource or "cost" associated with testability.

4.2.5.2 Choice of Objective and Constraint Functions

In general, although an allocation function may be analytically developed, either the objective function, the constraints or both will require the input of experientially developed data and/or experience driven approximations to complete the problem set-up in a solvable form. The use of heuristics (intuition) and the application of historically derived data is necessary. There are a number of sources which either provide data for or approaches to testability optimization. These include:

- MIL-HDBK 217 (Failure Rate Data)
- MIL-HDBK 472 (Repair Time Data)
- MATE GUIDE (G3V3P2 Appendix E)-Testability Overhead Estimation
- MIL-STD-2165 (Appendix B)-Inherent Testability Checklist
- RADC-TR-89-209 VOL II CADBIT (LIBRARY PACKAGE).

This list is far from complete. Furthermore, the historical data bases contained therein need to be expanded and periodically updated. Of particular interest is the MATE GUIDE G3V3P2 Appendix E formulation for testability overhead burden estimation.

The selection of objective and constraint functions, i.e. f(i) and g(ij) is based on the analysis made in the introduction of this section and thus depend on various criteria such as:

4.2.5.2.1 Failure Rates

4.2.5.2.2 Mission/System performance requirements

4.2.5.2.3 Measures of effectiveness of test systems.

4.2.5.2.4 LCC

4.2.5.2.1 Failure Rates

Failure rates may be used as weights (i.e. f(i) and g(ij)) in the objective and/or constraints functions. Failure rates are calculated to be the reciprocal of the Mean Time Between Failure (MTBF). For subsystems with high failure rates, it is expected that more reconfiguration and maintenance are required. Consequently, more fault coverage is needed for these subsystems. Thus, the problem can be mathematically stated as:

$$\text{max (or min)} \qquad \sum_{i=1}^{n} \left(\lambda_{Oi} / \lambda_S \right) TFOM_i \qquad (4\text{-}94)$$

where:

λ_{Oi} = failure rate of the i^{th} subsystem

λ_S = total failure rate of the system, i. e. $\sum_{i=1}^{n} \lambda_{Oi}$

$TFOM_i$ = coverage factors for i^{th} subsystem to be allocated.

4.2.5.2.2 Mission/System Performance Requirements

These requirements provide relationships between the mission scenario and system design criteria in which testability plays a part. The following is an example of such functions:

A. Mission Success, or Mission Failure Probability

B. Mission Criticality, Hazard Risk Index

C. Availability, Operational Readiness

A. Mission Success, or Mission Failure Probability

Mission failure percentages may be used as weights in the objective functions. Unlike the failure rate, the smaller the individual subsystem mission failure probability is, the less testability is needed for the subsystems. Thus, the problem formulation is:

$$\text{min (or max)} \quad \sum_{i=1}^{n} P_i * TFOM_i \tag{4-95}$$

where:

$$P_i = 1 - \exp(-\lambda_{Oi} t) \tag{4-96}$$

t is operating/mission time.

B. Mission Criticality, Hazard Risk Index

The Mission criticality or Hazard risk index provide a weighted objective function based on the frequency and criticality of the subsystem faults. Higher values of the hazard risk index indicate that the risk is low. In this case, the problem is formulated as follows:

$$\text{min (or max)} \quad \sum_{i=1}^{n} (10 - h_i) * TFOM_i \tag{4-97}$$

where h_i(frequency, criticality) = constant for each subsystem on the scale

usually from 1 to 10. (Equation 4-97 must be adjusted if the hazard rate scale is not from 1 to 10.)

NOTE: Equations (4-94), (4-95) and (4-97) are valid for each of the recommended TFOM ,that is, FD, FA, T_D, FI_P, FP, T_I, or for the new developed ones as defined in section 4.2.2. Furthermore, the maximization or minimization will depend on the choice of the TFOM.

C. Operational Readiness

The relationship tying the Operational readiness of a weapon system to the Fault detection is given by equation (4-36) or (4-41). This equation is interesting in the sense that it is directly related to the two TFOMs i.e., Fault detection (FD or Pr(D)), and False alarm (FA or Pr(FA)). The Operational readiness can also be used as the objective function to be maximized and is given by:

$$P_{or} = R(T_m) + Pr(D) * Pr(T_r \leq T_c) * Q(T_m) \quad (4\text{-}98)$$

or

$$P_{or} = \{ (1 - Pr(FA)) + [Pr(FA) * Pr(T_{FA} \leq T_c | \text{ false alarm })] \} R(T_m) + [Pr(D) * Pr(T_r \leq T_c) * Q(T_m)] \quad (4\text{-}99)$$

The constraints can also be the same or similar at all levels of indenture. These may include data from failure rates, overhead burdens (weight, volume, power, cooling, etc.), risk and criticality factors, mission failure probability or cost restraints. In addition, upper and lower bounds (ub and lb) for each TFOM should be included as a constraint in the calculations. In most cases, the constraints are of the form:

$$lb_i \leq TFOM_i \leq ub_i \quad (4\text{-}100)$$

$$\sum_{i=1}^{n} constr(i,j) * TFOM_i \ (\geq \text{ or } \leq) \ C \quad (4\text{-}101)$$

where C is the system constraint requirement, and constr(i,j) are the various weighting factors for the constraints, with j(j = 1,.....m) being the index representing the j^{th} constraint.

For example, the failure-rate weighted constraints can be expressed as:

$$\sum_{i=1}^{n} (\lambda_{Oi} / \lambda_S) \, TFOM_i \geq C \qquad (4\text{-}102)$$

where C is the system level testability requirement.
Note also that, on the assumption that operational readiness, mission reliability, scheduled time between missions and maintainability are known or given, it is possible to rewrite equation (4-98) as:

$$Pr(D) = (P_{or} - R(T_m))/ (Pr(T_r \leq T_c) * Q(T_m)) \qquad (4\text{-}103)$$

This equality is really an inequality since, operational readiness is a level to be achieved, hence equation (4-103) could be rewritten as:

$$Pr(D) \leq (P_{or} - R(T_m))/ (Pr(T_r \leq T_c) * Q(T_m)) \qquad (4\text{-}104)$$

Thus, a constraint on detectability or fault detection is also provided using equation (4-103) or (4-104). A similar bound may be derived from equation (4-99).
At the subsystem level, we note that allocation to the elements of subsystems can be made based on maintenance (fault isolation to the LRU at the flightline) as well as system considerations. Thus, in addition to the weighting factors used in the choice of "cost" functions, the following may also be important from the maintenance standpoint:

- reliability importance of LRU in the overall reliability block diagram.
- cost to implement LRU diagnostics or coverage
- level of technology.

The assumption made when using the level of technology as a constraint is that the average testability for VHSIC technology is to be greater than the average for digital, which in turn is greater than the average for hybrid, which in turn is greater than the average for analog.

4.2.5.2.3 Measures of Effectiveness of Test Systems

We observed in the introduction of this section that, DFT is best achieved by minimizing false flight line removals, RTOK or CND events, and maintain low false alarms at the O-level. In implementing diagnostic systems, four type of problems arise (see section 4.2.2):

- False alarms
- CND
- RTOK
- Failure to diagnose

In order to assess the capability of a diagnostic system at any level of repair, and in particular at the O-level, the following measures of effectiveness were derived from the system requirements:

A. False Removal (FRMV)

B. Failure to Diagnose (FDG)

C. False Alarm Correction (FAC)

D. Expected Number of Removals per Failure (E(RMV))

These measures of effectiveness which can be used as metrics for the allocation, were shown to be functions of more than one TFOM, hence non-separable.

A. False Removal

This is a function of false alarm and false isolation as given by equation (4-9). The problem can then be stated as:

$$\min FRMV = [1-Pr(F)] * Pr(FA) * Pr(I \mid FA) \qquad (4\text{-}105)$$

Minimizing false removal will lead to maintaining low false alarm.

B. Failure to Diagnose

This represents the capability of correct diagnosis, and is a function of the FD/FI of a system as given by equation (4-10). The objective function to be minimized is:

$$\min (FDG_i) = Pr(F) *[1 - Pr(FD) * Pr(FI_i)] \quad (4\text{-}106)$$

These measures represent the accuracy of the test system and its ability to perform according to the requirements. FRMV and FDG are both between 0 and 1, and any value in this range will indicate that the test system is effective.

C. False Alarm Correction

This measure is the CND and the RTOK at different levels, as was shown in section 4.2.2.3.3. Using equation (4-11), we get

$$FAC = 1 - Pr(I \mid FA) \quad (4\text{-}107)$$

Substituting the relationships of equation (4-6) yields:

$$FAC = Pr(CND) \quad (4\text{-}108)$$

Thus, CND's (due to false alarms) could be used as the objective function to be minimized. Note that FAC is also between 0 and 1, however a small value may indicate an inferior test system capability.

D. Expected Number of Removals per Failure

This measure is built into the Fraction of false pulls TFOM (FP) defined in section 2.0. Isolation of a failure to one unit or LRU is ideal in the sense that it will result in one removal per failure assuming 0% false alarm. Hence, we would like E(RMV) to be 1,that is, one maintenance action per failure, or at least minimum. This choice of E(RMV) as an objective function leads to a conflict, since the time to isolate may be larger. Thus, the E(RMV) measure derived from test system requirements such as FD, FI, and erroneous fault indications, and given by equations (4-16)-(4-19), is the objective function to be minimized, subject to the minimum limit on the time to isolate. The problem is then stated as:

$$\min E(RMV) = \min \sum_{i=1}^{n} \lambda_i E(RMV)_i \qquad (4\text{-}109)$$

where:

$$E(RMV) = Pr(FD)_O * RMVFD + (1 - Pr(FD)_O) * RMVNFD + \lambda_{FA} * RMVFA \qquad (4\text{-}110)$$

λ_{FA} is the number of false alarm occurrences per system failure. Note that a value of E(RMV) between 0 and 1 will indicate that the test system is effective.

4.2.5.2.4 LCC

There are several choices for objective and constraint functions and some of the choices conflict. In wartime, the criteria is to maintain a high probability of mission success, however, in peacetime, the criteria is LCC which is largely determined by maintenance costs. This section shows the various forms of "cost" functions related to LCC.

A. Acquisition Costs

B. Design Costs

C. Costs Associated with the Measures of Effectiveness

A. Acquisition Costs

These costs may be used as weights in the objective and constraints functions. The objective function to be minimized is given by:

$$\min \quad \sum_{i=1}^{n} c_i * TFOM_i \qquad (4\text{-}111)$$

where c_i are overhead burdens due to testability in terms of acquisition costs for each subsystem.

Constraints used for this example include the maximum limits on the TFOMs, and the failure-rate weighted constraint given by equations (4-100)-(4-101).

B. Design Costs

The design costs as described in section 4.2.4.1.4 (d) are exponentially related to the percentages of fault detection, FD, and given by:

$$C_D = \exp(a * FD) * C_u \qquad (4\text{-}112)$$

where:
a is a function of the state-of-the-art in terms of design engineering, and a value between 0 and 1 is desirable.
C_u is the cost of a unit (subsystem, LRU, etc.)
The objective function to be minimized is given by:

$$\min \quad \sum_{i=1}^{n} C_{Di} = \sum_{i=1}^{n} \exp(a * FD_i) * C_{ui} \qquad (4\text{-}113)$$

Equation (4-113) shows the design cost as non-linear objective function of FD, but the problem has still a separable structure.

The constraints used in this case are the maximum limit on FD, and may be the failure-rate weighted together with other overhead burden requirements normalized with respect to FD.

C. Costs Associated with the Measures of Effectiveness

The objective function in this case could be either the costs associated with each measure of effectiveness, that is, cost of false removal (C_{FRMV}), cost of failure to diagnose (C_{FDG}), or cost of false alarm correction (C_{FAC}), given by equations (4-48) - (4-50), or a mean cost of the diagnostic system and its malfunctions. This mean cost is used to evaluate and compare the capability of various diagnostic systems. It can discriminate between different test systems based not only on their costs or FD/FI capability, but also on the mean cost of the burden of their imperfections during their life. This measure considers also false alarm and false removals and their cost. The objective function to be minimized is given by:

$$\min LCC = \min [C_{PS} + \mu (C_{FRMV} + C_{FDG} + C_{FAC})] \qquad (4\text{-}114)$$

where:

C_{PS} is the prime system cost.

μ is the mean number of missions during the life cycle of the diagnostic system.

The different costs built in equation (4-114) are spares costs, mission related costs, and related maintenance costs such as:

- cost of removing and replacing the unit
- cost of isolating the unit
- cost of transporting the unit to the next level (D-level)
- cost of having a good unit at the next level (D-level).

Another measure of interest is the cost associated with the expected number of removals per failure, E(RMV). This measure was shown to be relatable to the O & S costs for maintenance manpower and spares. The more effective the test system (i. e., the lower the value of E(RMV)), the less manpower and spares will be required to support the system. Thus, the maintenance manpower costs, CMMP, and spares cost, CSP, as defined by equations (4-51), and (4-56) can be used as objective/ constraint functions. Minimizing these costs will lead to an optimal and cost effective allocation of the testability requirements.

Summary

The selection of "cost" functions included a variety of mathematical forms in order to reveal the flexibility of the solution technique. The testability allocation methodology is based c.. ..e Augmented Lagrangian technique which treats system performance as functions of the TFOMs. As can be seen from the above analysis, there are several choices and combinations for the objective and constraint functions, and the list is far from complete. These different choices will generally yield different allocations as shown in the examples presented in the next section.

4.2.5.3 Application: Example Problems

The TAM algorithm was tested successfully on representative problems from reference [21]. The TFOM considered is FD/FI of subsystems for BIT coverage. The objective functions are weighted according to failure rates and mission failure probability.

The TAM is used at both the system level (allocating to subsystems) and at the subsystem level (allocating to elements of subsystem). Using the MATE curves (see Appendix A) the relationships between the overhead burden requirements (i. e., weight) and the FD/FI are derived. These curves are shown in Figure E-4 and are smooth. The problem is made separable by fitting piecewise linear approximations to each of the required curves. Tables T-9 and T-10 contain the data for subsystem parameter allocation, and the results of sample runs on the two examples respectively.

	MTBF	FAILURE RATE	MISSION FAILURE PROBABILITY
SUBSYSTEM 1	2500	.000400	.0010
SUBSYSTEM 2	1250	.000800	.0020
SUBSYSTEM 3	588	.001701	.0030
SUBSYSTEM 4	133	.007519	.0150
SUBSYSTEM 5	278	.003597	.0080
SUBSYSTEM 6	3750	.000267	.0010

TABLE T-9 SUBSYSTEM PARAMETER ALLOCATION

4.2.5.3.1 System Level

A. Maximizing Failure Rate

Using the table of failure rates of subsystems (Table T-9), and the graph of the weight burden vs FD/FI (Figure E-4), the percentage of fault coverage can be calculated. The problem is stated as follows:

$$\max(.0004x_1 + .0008x_2 + .001701x_3 + .007519x_4 + .003597x_5 + .000267x_6).$$

Stating it as a minimization problem:

$$\min -(.0004x_1 + .0008x_2 + .001701x_3 + .007519x_4 + .003597x_5 + .000267x_6). \quad (4\text{-}115)$$

Constraints used in this example include the maximum limit on FD/FI, and the weight burden normalized with respect to the probability of isolation. The constraints are:

$$0 \leq x_i \leq .99 \text{ for } i = 1, 2, 3, \ldots 6$$

and

$$428.18x_1 + 124.09x_2 + 6.7x_3 + 14.67x_4 + 75.82x_5 + 24.55x_6 - 500.77 - 140 \leq 0. \quad (4\text{-}116)$$

The results indicate (see Table T-10) that subsystem 2,3,4,5 and 6 required maximum amount of fault coverage. Subsystem 1 required slightly less coverage.

OBJECTIVE FUNCTION	SUBSYSTEMS					
	1	2	3	4	5	6
MAXIMIZING FAILURE RATE ALLOCATION	0.928	0.99	0.99	0.99	0.99	0.99
MINIMIZING PROBABILITY OF MISSION FAILURE	0.931	0.99	0.99	0.99	0.99	0.00

TABLE T-10 SUMMARY OF SAMPLE RUN RESULTS

B. Minimizing Mission Failure Probability

Using Table T-9 for the mission failure probability data, the objective function to be minimized is:

$$\min(.001x_1 + .002x_2 + .003x_3 + .015x_4 + .008x_5 + .001x_6) \quad (4\text{-}117)$$

The constraints include:

$$0 \leq x_i \leq .99 \text{ for } i = 1, 2, 3, \ldots 6$$

$$428.18x_1 + 124.09x_2 + 6.7x_3 + 14.67x_4 + 75.82x_5 + 24.55x_6 - 500.77 - 140 \leq 0.$$

and

$$-(.0004x_1 + .0008x_2 + .001701x_3 + .007519x_4 + .003597x_5 + .000267x_6 - .97 * .01428) \leq 0 \quad (4\text{-}118)$$

The last two constraints are a weight burden constraint and the failure rate weighted constraint for a system level FD/FI of 0.97.

The results indicate that subsystem 1, 3, 4 and 5 require the maximum amount of fault coverage. Subsystem 2 requires slightly less coverage while subsystem 6 requires no coverage.

4.2.5.3.2 Subsystem level

The allocation of subsystem level testability requirements to the LRU level is shown here. Table T-11 contains data for LRU parameter allocation. There are 11 modules whose technology is a combination of VHSIC, Digital, hybrid between analog and digital, and analog.

Modules	Technology	FAILURE RATE $/10^{6}$ hrs
MAS1	VHSIC	1.000000
MAMB1	VHSIC	1.000000
CSP1	VHSIC	6.097560
VIDEO SWITCH	DIGITAL	20.000000
SIU1	A & D	16.129032
ESIU1	VHSIC	19.193858
PI BUS 1	ANALOG	1.060445
DN1	DIGITAL	2.500000
DM1	DIGITAL	45.454545
CRT(HUD)	ANALOG	45.454545
H-VOLT POWER	ANALOG	28.571429

TABLE T-11 LRU PARAMETER ALLOCATION DATA

The objective function to be minimized is

$$\min \quad \sum_{i=1}^{11} x_i \qquad (4\text{-}119)$$

where all the weights are assumed to be 1.

The importance of FD/FI testability to the module technology is a constraint in this case. It consists of three parts which are defined as follows:

- average testability for VHSIC modules ≥ average for digital
- average testability for digital modules ≥ average for A & D
- average testability for A & D modules ≥ average for A only

Thus, using the data in Table T-11 and the above reasoning, we get:

$$.25(x_1 + x_2 + x_3 + x_6) - (1/3)(x_4 + x_8 + x_9) \geq 0 \quad (4\text{-}120)$$

$$(1/3)(x_4 + x_8 + x_9) - x_5 \geq 0 \quad (4\text{-}121)$$

$$x_5 - (1/3)(x_7 + x_{10} + x_{11}) \geq 0 \quad (4\text{-}122)$$

The maximum limit on FD/FI and the failure-rate weighted constraint vary with each sample run as shown in Table T-12.

RUNS	SYSTEM REQUIREMENTS	LOWER BOUND	UPPER BOUND
1	0.97	0.00	1.00
2	0.94	0.00	1.00
3	0.90	0.00	1.00
4	0.97	0.00	0.99
5	0.97	0.50	0.99
6	0.99	0.00	1.00
7	0.97	0.00	1.00

TABLE T-12 SAMPLE RUN REQUIREMENTS

The results as shown in Table T-13, are identical to the ones found in [21]. It should be noted that, the values from run 7 were omitted, since all the x_i are equal to 0.97.

Modules	RUNS					
	1	2	3	4	5	6
1	0.237648	0.000000	0.000000	0.539805	0.500000	0.865558
2	1.000000	0.621864	0.442937	0.990000	0.947279	1.000000
3	1.000000	1.000000	1.000000	0.990000	0.990000	1.000000
4	1.000000	0.966398	0.832202	0.990000	0.990000	1.000000
5	0.809412	0.655466	0.610734	0.877451	0.856820	0.966390
6	1.000000	1.000000	1.000000	0.990000	0.990000	1.000000
7	0.000000	0.000000	0.000000	0.000000	0.500000	0.000000
8	0.428236	0.000000	0.000000	0.652353	0.590459	0.899168
9	1.000000	1.000000	1.000000	0.990000	0.990000	1.000000
10	1.000000	1.000000	1.000000	0.990000	0.990000	1.000000
11	1.000000	0.966398	0.832202	0.990000	0.990000	1.000000

TABLE T-13 TAM RESULTS: SAMPLE LRU ALLOCATIONS

4.2.5.4 Summary

The TAM algorithm ran satisfactorily on all examples, at the system and subsystem levels. Maximizing the failure-rate-weighted FD/FI coverage results in a different allocation from mission-related criteria. In this latter case, the TAM produced a different and slightly better solution than the one in [21]. At the subsystem level (Table T -13), and for the other examples (Table T-10) the results were identical.

5.0 BOTTOM-UP BIT PRIORITIZATION

The objectives of BIT prioritization from a bottom-up perspective is , given a detailed design of the functional portion of a system, to evaluate potential diagnostic tests and assess the degree to which they should be incorporated in a BIT subsystem. In order to accomplish this we need a basis for that evaluation. It is the intent of this discussion to establish that basis and develop a corresponding prioritization approach. This section of the report is organized in three parts. First, underlying assumptions and pertinent definitions are presented to set the stage. Second, the objectives of BIT prioritization are formulated in terms of those assumptions and definitions. Finally, an approach is developed that satisfies those objectives.

5.1 Assumptions and Definitions.

In the bottom-up perspective, we mandate that a detailed design of the functional portion of a system exist. This typically includes information such as specific components, wiring diagrams, and possibly assembly drawings. In addition, the operational characteristics and prescribed maintenance philosophy for the system must have been defined.

We now constrain the problem with a number of simplifying assumptions.

Assumption 1: BIT may be used for fault detection, fault isolation, or some combination of the two. The specific role designated will be considered the mission of BIT.

Assumption 2: A diagnostic test may be performed by one of three classes of systems: BIT, external Automatic Test Equipment (ATE), and manual testing methods (which may or may not require test equipment).

Assumption 3: Once a mission has been identified for BIT, no part of that mission will be performed by alternate types of test systems (i.e., ATE or manual testing). This assumption limits the scope of our prioritization by not having to consider tradeoffs involving mixed strategies, such as carrying out part of the diagnostic mission in BIT and part using ATE.

Assumption 4: The cost of performing, not of implementing, any test in BIT is substantially less (approximately by an order of magnitude, derived using the 10-to-1 rule for testing during manufacturing) than by other means.

Assumption 5: The time required to perform an individual test incorporated in BIT is substantially less than by other means. This is due to BIT's lack of a need for setup time and its rapid execution speed.

Assumption 6: The costs, in terms of power and consumables, for tests in BIT are estimable.

Assumption 7: The time required to run each test within a BIT subsystem is estimable.

Assumption 8: The cost of implementation of each test in BIT is estimable.

We now define pertinent terms associated with the bottom-up prioritization subtask.

System - A group of interconnected elements, sometimes called components, that performs some function or functions. The use of the term "system" here is not associated with any formal level of indenture. For the purposes of this discussion, a system may be an entire weapon system, an aircraft LRU, or even a circuit board.

Component - A constituent element, many of which comprise a system. The use of the term "component" here is not associated with any formal level of indenture. For the purposes of this discussion, a component is the next level of indenture lower than a system.

Aspect - A specific functional characteristic or constraint that, if violated, constitutes a failure. An aspect models two phenomena, internal failure modes and potential system input errors. Aspects are functional abstractions and are not restricted to any specific level of hardware indenture (e.g., components or subsystems).

Test - A procedure in which some stimulus is provided and/or known a priori, and a related response is observed. If the response is expected, the test is said to pass; otherwise it is said to fail. One test, T_i, may employ numerous test points.

$$T_i: [TP_j, TP_k, \ldots, TP_m]$$

Test Point - A physical location in space and/or time where tests may be performed. One test point, TP_i, may be used by many tests.

$$TP_i: [T_j, T_k, ..., T_m]$$

Dependency - A logical relation that may exist between two aspects, two tests, or an aspect and a test. An element (i.e., test or aspect) is said to depend on another element if the failed condition of the first implies the incorrect performance of the latter, and the correct performance of the latter implies the correct performance of the former. For example, if the test T_i depends on aspect A_j, then a failure that occurs, which is characterized by A_j, implies that test T_i will fail. Conversely, if T_i passes, then A_j is not failed.

Process Test - A test that, if passed, validates the correctness of a process parameter of a given host system. That process parameter may depend on numerous other process tests and aspects. Similarly, numerous other process tests and aspects may depend on that system parameter. An example of this type of test might be the measurement of a power supply's output voltage.

Component Test - A test that validates the goodness of a specific aspect(s) in functional isolation from its host system. These tests have no dependencies other than that aspect(s). An example of this type of test might be the use of built-in self test (BIST) capability on a VHSIC device.

5.2 Formulation of Objectives.

In the environment described by the above definitions and assumptions, we have a system design that, in general, does not as yet contain any tests.

The first objective in our prioritization must necessarily be to identify potential diagnostic tests for incorporation in a BIT subsystem Prioritization implies some type selection (based on the results of the prioritization). It is therefore reasonable to expect that we choose more tests than required. As such, the test selection process should attempt, in a systematic manner, to identify an exhaustive set of candidate BIT tests for subsequent prioritization, selection, and implementation in BIT.

The objectives of the bottom-up BIT prioritization task are three-fold. First, we must identify candidate tests for incorporation in BIT. The resulting set of tests is defined as:

$$\xi_C = [\, T_1, T_2, T_3, \ldots, T_n]$$

Given the set ξ_C, we then must select a subset ξ_S, that is optional in some sense. The sense in which ξ_S is optimal should be based upon the predetermined mission of the BIT subsystem. Typically we want to select those tests from ξ_C that allow BIT to perform its intended mission as rapidly as possible while minimizing implementation costs, and its operational resource requirements.

Finally, it is necessary to score the individual tests within the subset ξ_S in terms of their applicability to BIT. Thus, if one were forced to choose a subset of ξ_S, say ξ_{SS} , then we would want to choose those tests that are most applicable to BIT (i.e. have the highest scores).

Thus far we have identified three goals that together constitute the bottom-up BIT prioritization objective. They are:

- Identify set ξ_C of potential BIT tests
- Select an optimal subset ξ_S
- Rank the members of ξ_S

In the following section an approach is described that address these goals.

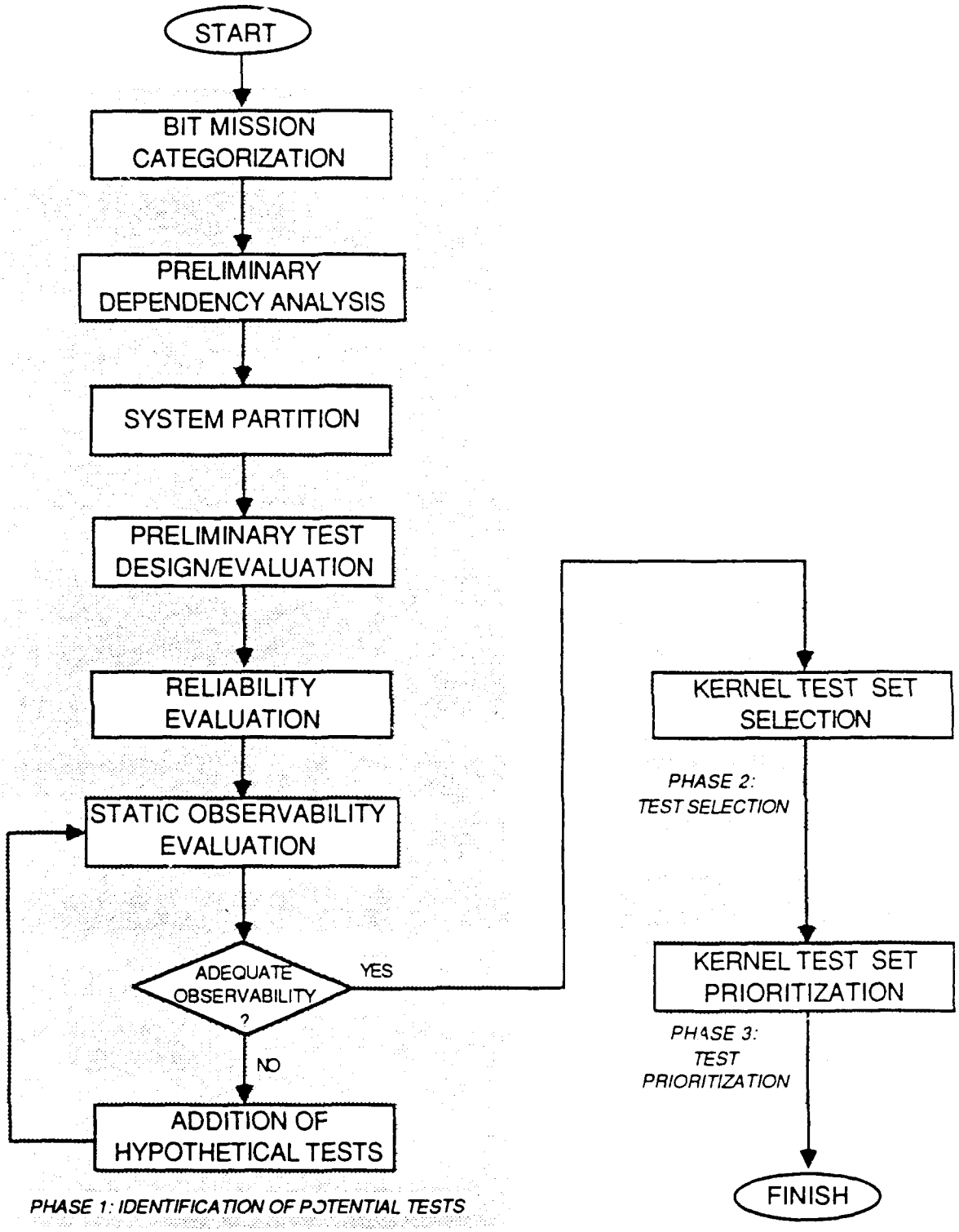


Figure 5.3-1 Flow Diagram for Bottom-Up BIT Prioritization Procedure

5.3 Approach.

The prescribed approach for a bottom-up BIT prioritization has a phase for each of the three aforementioned goals. The first phase is the identification of the potential tests (i.e., set ξ_C). The second phase results in the selection of an optimum subset of tests ξ_S, and the third phase involves the ranking of the elements in ξ_S. The overall process is depicted in Figure 5.3-1. All of those phases are described in the following paragraphs. Throughout the remainder of this discussion the sample system of Figure 5.3-2 will be used for purposes of illustration.

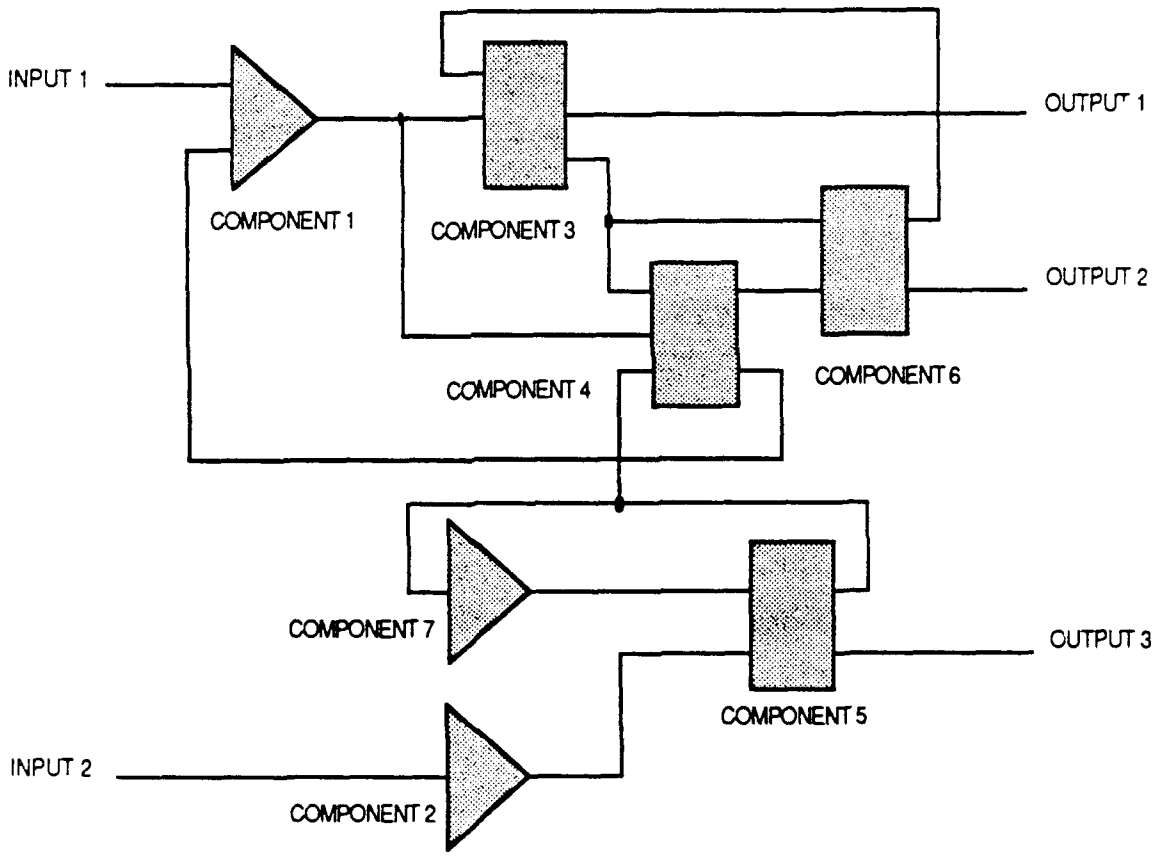


Figure 5.3-2 A sample system for discussions on a bottom-up approach to BIT prioritization. The system consists of 7 components, 2 inputs, and 3 outputs. The nature of the technology employed is unspecified.

5.3.1 Phase 1: Identification of Potential Tests.

The overall objective of Phase 1 is to identify all the potential diagnostic tests for the system under evaluation and determine their pertinent relationships. This phase has seven steps (Figure 5.3-1): BIT mission categorization, preliminary dependency analysis, system partition, preliminary test design/evaluation, reliability evaluation, static observability evaluation, and addition of hypothetical tests.

5.3.1.1 BIT Mission Categorization.

The test selection and prioritization must account for the role or mission of the BIT subsystem. According to Pliska et al. [1], BIT may perform some combination of two fundamental tasks, fault detection and/or isolation. For the purposes of this discussion, we will categorize BIT as belonging exclusively to one of two classes based upon its intended role, "Detection BIT" or "Isolation BIT". As will be discussed later, this restriction may be relaxed in order to permit a hybrid role involving elements of both (e.g., fault detection with isolation of critical faults).

5.3.1.2 Preliminary Dependency Analysis.

The dependency analysis is carried out on the system under design in accordance with techniques spelled out in [2] and [3]. First, in a component-simulation-based approach [4], the components that comprise a system are analyzed to determine their functional/failure modes (i.e., aspects) as shown in Figure 5.3-3. These aspects become nodes in a directed dependency graph (see Figure 5.3-4).

The topology formed by the interdependent aspects is then used to identify all hypothetical points of observation (i.e., process tests). Every dependency between aspects is associated with a potential BIT diagnostic test. Process tests are typically used here because of their high information yields. However, deemed appropriate, component tests may also be incorporated. For example, if we are considering a VHSIC microprocessor device, we are likely to want to make use of BIST rather than attempting to identify functional tests for it. All of these tests are then incorporated in the dependency graph. The completed graph models the dependencies between different function/failure-mode aspects, between the various potential process tests, and between aspects and tests.

5.3.1.3 System Partition.

The third step in this phase is the partitioning of the dependency graph If the BIT mission class is strictly fault detection, the partition follows (Figure 5.3-5 demonstrates the detection partition for our sample system):

1) The existing system in its entirety is considered as a replaceable unit whose aggregate failure rate is equivalent to the sum of the failure rates of its constituent components

2) A virtual replaceable unit is declared that includes an aspect which represents the no fault condition and is assigned a likelihood of occurrence based on the failure rate of the system components

3) Each system input is regarded as an aspect contained in the "no fault" virtual replaceable unit that can fail at rates determined by estimation techniques or, when available, field assessment

If, on the other hand, the BIT mission is strictly isolation, the partition is as follows (Figure 5.3-6 shows our sample system partitions for Isolation BIT):

1) Each constituent system component is regarded as a replaceable unit with failure rates as determined in Phase I

2) Each system input is regarded as a replaceable unit that can be isolated with failure rates are determined by estimation or using field data, when available

3) A virtual replaceable unit and associated aspect are created to account for the no fault condition with occurrence probability based upon some specification for allowable CND (Cannot Duplicate Event) or RTOK (Retest OK) rates

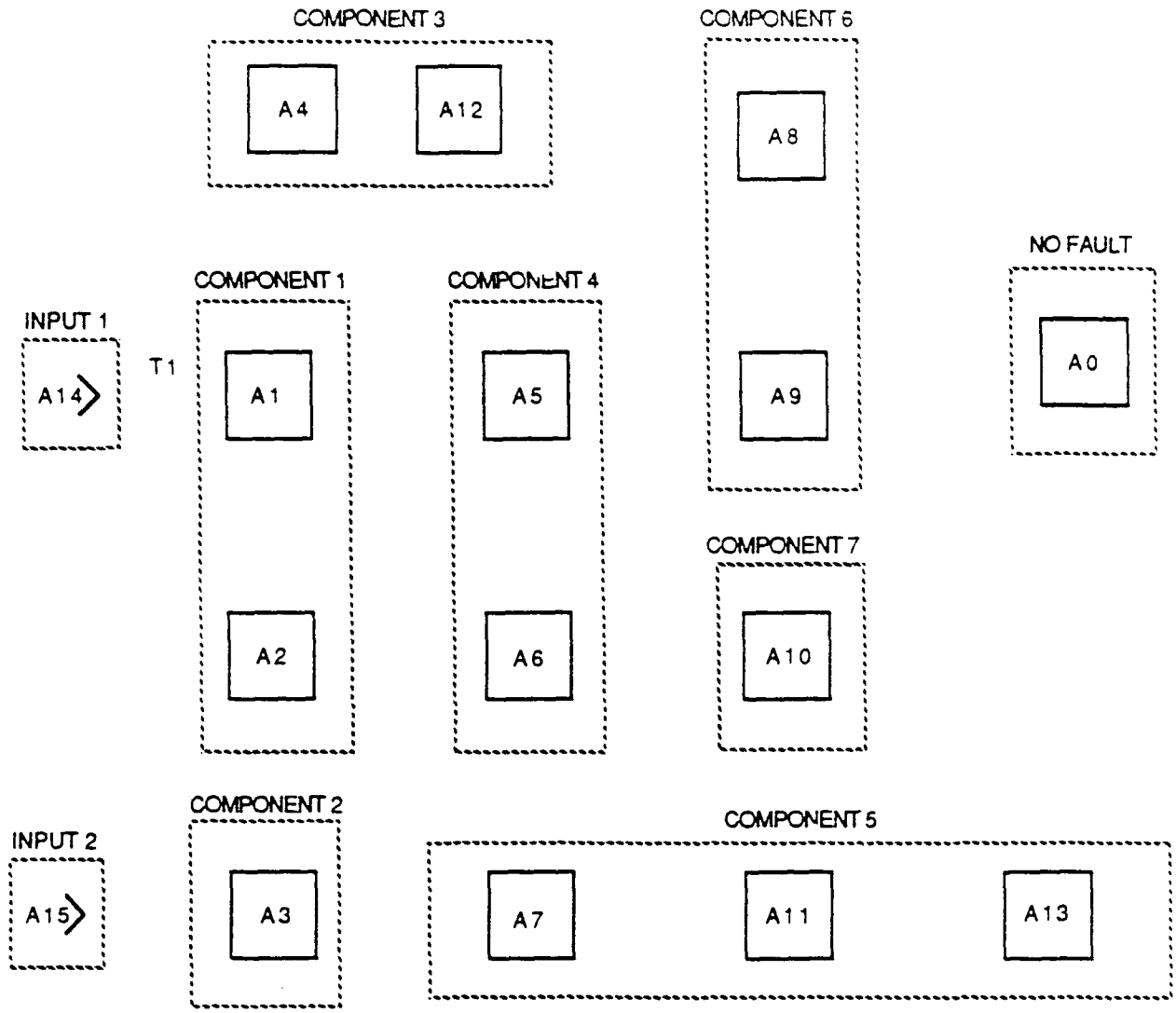


Figure 5.3-3 The functional/failure-mode aspects for the components and inputs of sample system (from Figure 5.3-2).

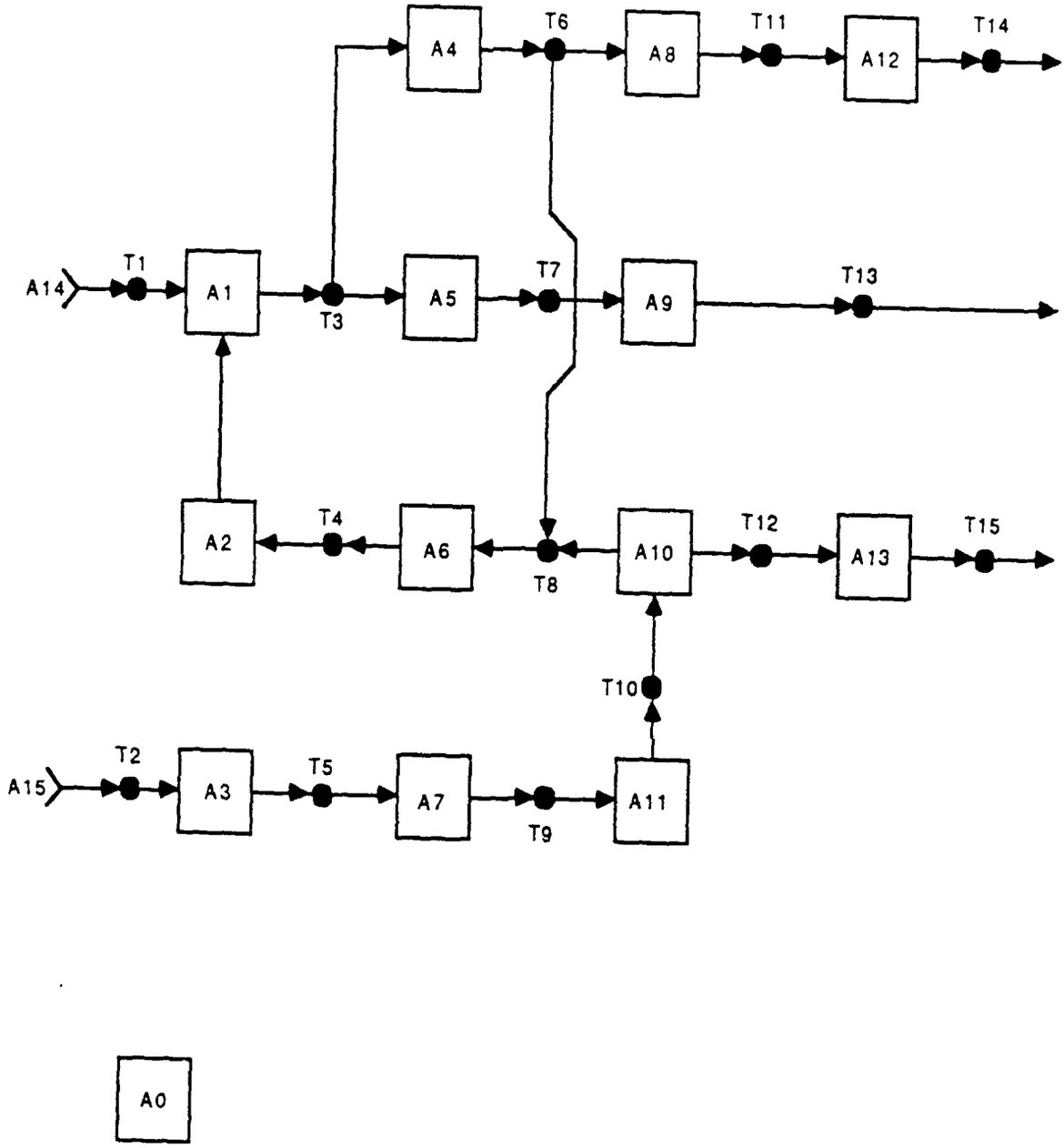


Figure 5.3-4 Dependency model for sample system (from Figure 5.3-2). The aspect A0 represents a "no fault" condition.

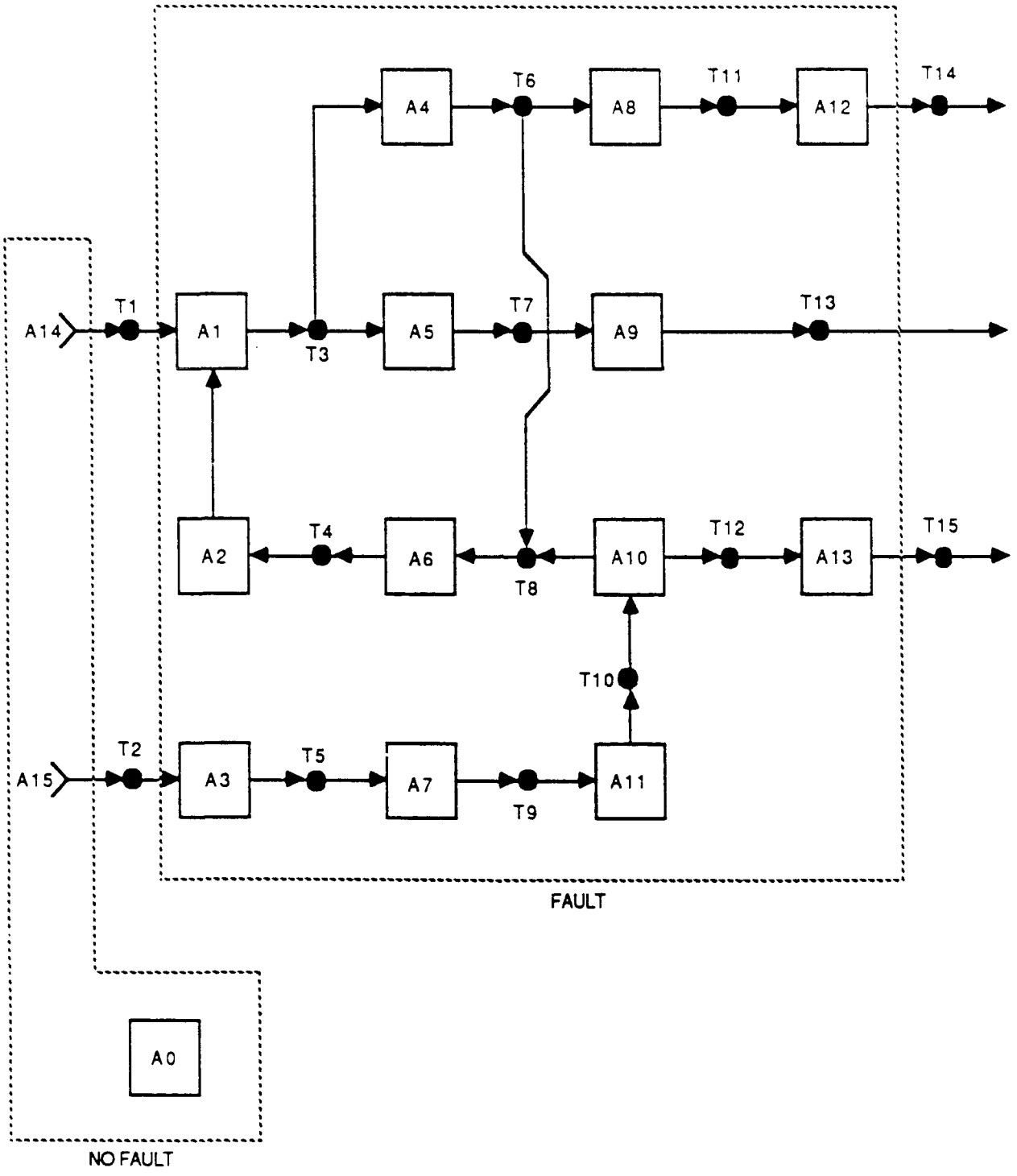


Figure 5.3-5 Sample system dependency model partitioned for fault detection. Notice that the input aspects A14 and A15 are contained in the "no fault" replaceable unit. If they were to fall within the "fault" replaceable unit, they would be a source of false alarms.

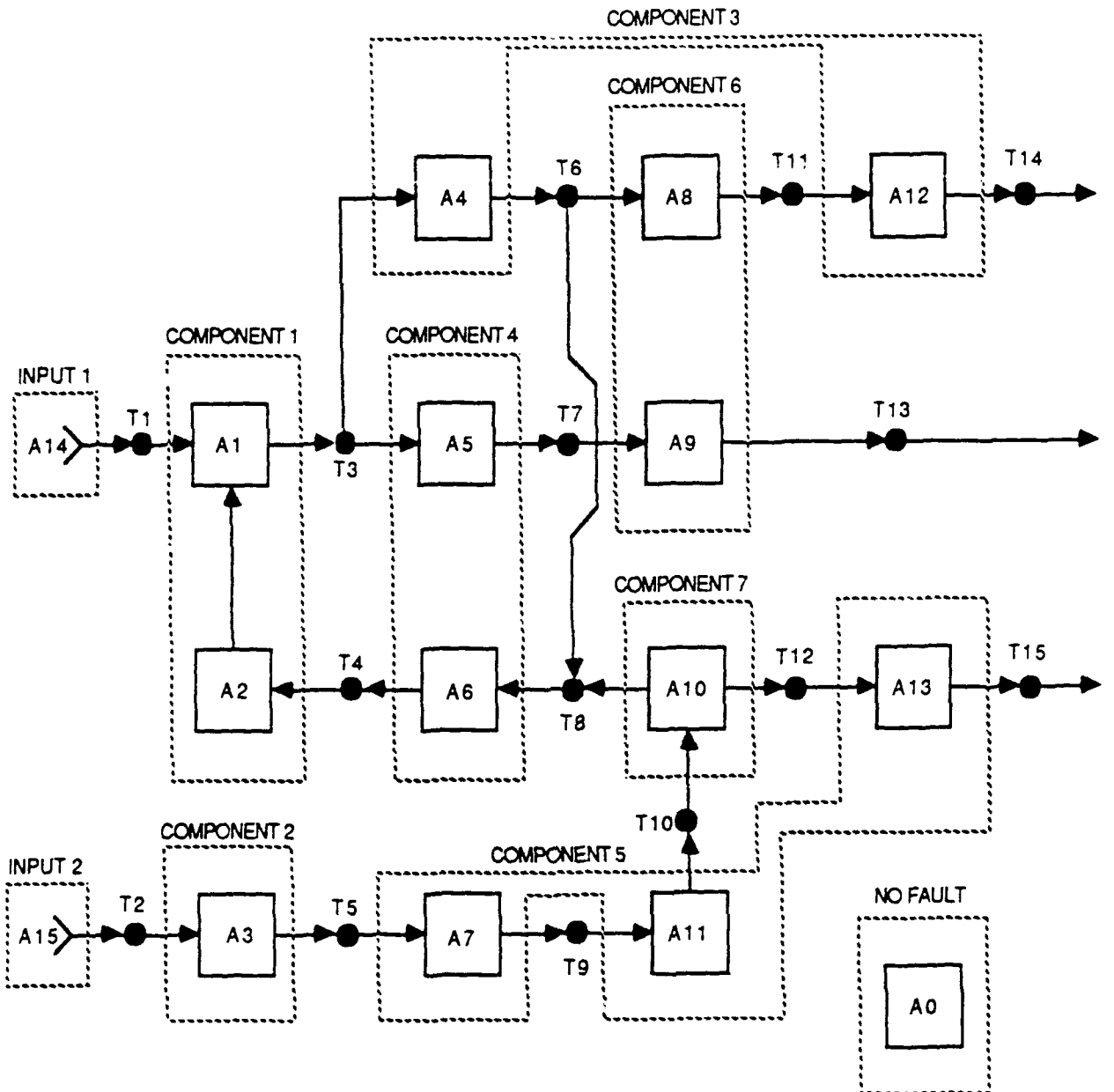


Figure 5.3-6 Sample system dependency model partitioned for fault isolation. Observe that components could be grouped in any arbitrary fashion to yield a hybrid detection/isolation BIT.

5.3.1.4 Test Design and Cost Evaluation.

The next step in Phase I, the implementation costs of the potential tests in ξ_C must be estimated. This may best be accomplished by involving experienced test engineers. Particular focus must be maintained on estimating the implementation costs, execution times and operational resources associated with BIT.

5.3.1.5 Reliability Evaluation.

The aspects of the graph must have their likelihoods (i.e., relative probabilities of failure) determined. This is usually accomplished by examining fielded reliability data for the components (such as those compiled by RADC) in combination with using techniques similar to those in MIL-HDBK-217. The relative likelihoods for the partitions are then computed by summing over the failure probabilities of the aspects that each partition contains.

5.3.1.6 Static Observability Evaluation and Addition of Hypothetical Tests.

At this point, a preliminary high level analysis of the test observability is made. A tool such as the IDSS/WSTA [2] or STAMP [3] may be used for this activity. Any ambiguities are identified here and can be rectified by the addition of component tests. An example of the changes to our sample system resulting from such an analysis are shown in Figure 5.3-7. It is quite important that a test engineer be involved in this step. The tests that are added must, in turn, be evaluated to estimate their implementation costs, execution times, and operational resources. In general, we desire to have substantially more tests in our candidate set ξ_C than are required for adequate observability. The larger the set ξ_C, the better will be the resulting selection of tests, ξ_S. Although tools such as WSTA or STAMP, in addition to allowing analysis of observability, may be used to advise on the pruning of the test set ξ_C, no tests should be eliminated at this point. This particular activity must be deferred until the prioritization takes place.

5.3.1 7 Results from Phase 1.

At the conclusion of Phase 1, an inherent testability analysis has been performed and a dependency model created. The dependency model contains the candidate test set ξ_C and the relationships they have within the system. They include the topological relationships between tests

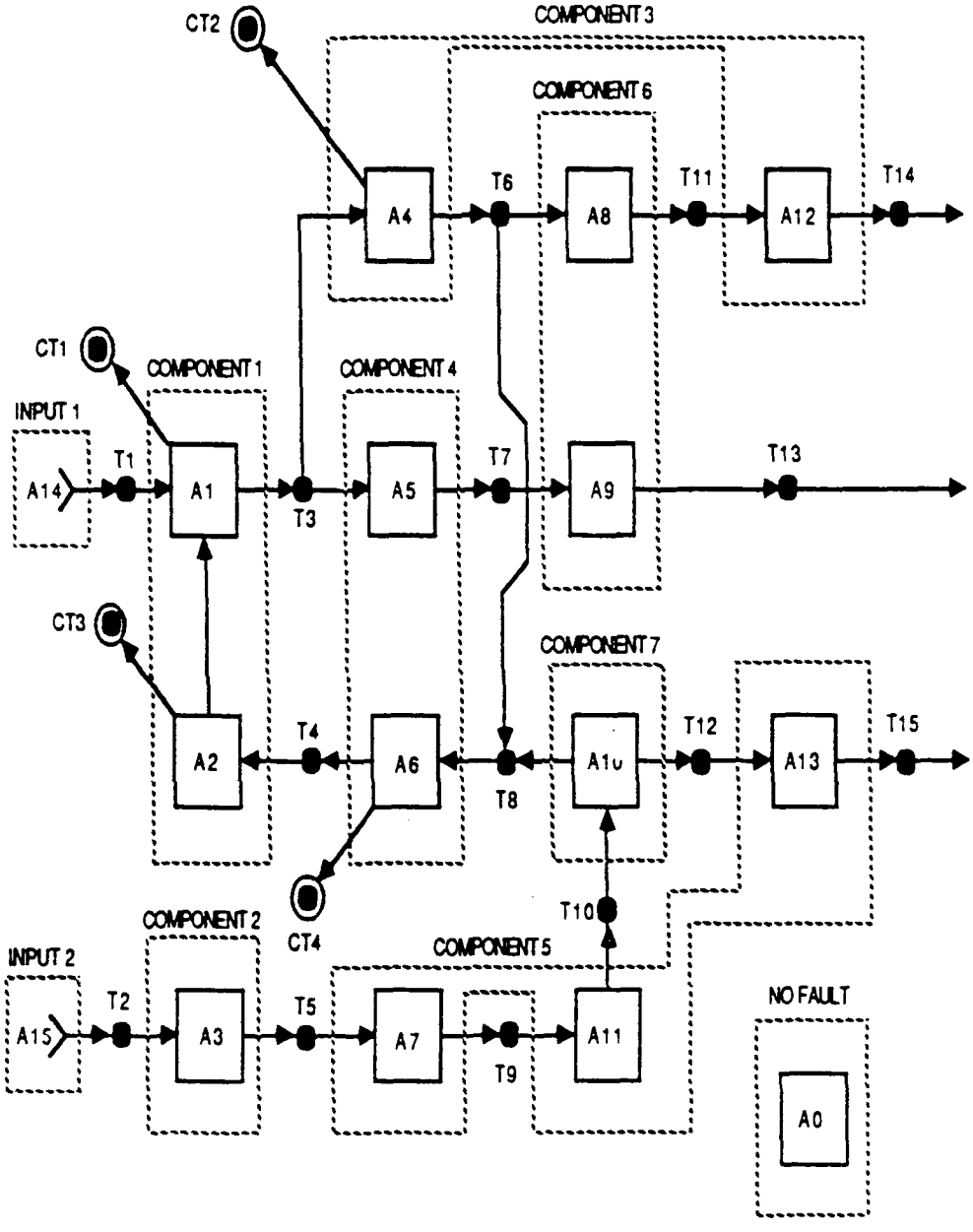


Figure 5.3-7 Modified isolation partition for sample system. Note that component tests CT1 through CT4 have been added. The preliminary testability analysis identified a feedback loop that created an ambiguity group containing aspects A1, A2, A4, and A6. The components tests were added to eliminate the ambiguity group. It turns out that only three of the four additional tests are necessary.

and other tests, as well as tests and aspects. Other than topological relationships, the relative likelihoods of aspects and robustness of the tests are modeled. The model is also partitioned to represent the conclusion space (e.g., a particular component is faulty versus the system is faulty) that is required by the BIT mission category. Finally, the dependency model contains the test implementation costs, execution times, and operational resource requirements, all with a focus on their implementation in BIT.

5.3.2 Phase 2: Test Selection.

The objective of the test selection process is to choose the optimum set ξ_S for implementation in, and use by, BIT. The selection of the test set ξ_S is accomplished by the construction of an optimal (or near-optimal) decision tree. During the construction of this decision tree, the bases available for test selection are their implementation/acquisition costs, execution resource requirements, and execution times. The cost of implementation for any given candidate test is a dynamic function. In the event that it has been selected for one node in the tree, its cost for use at another node is zero. The overall cost function, $\gamma(i)$, that is to be minimized is of the following form:

$$\gamma(i) = A_I C_{Ii} \delta_i + A_R C_{Ri} + A_T t_i \qquad (5\text{-}1)$$

where A_I , A_R , and A_T are importance weights established by the system designer, based upon BIT performance criteria. C_{Ii} is the estimated cost for implementing the i^{TH} test, and δ_i is 1 if test i has not yet been selected in the tree and 0 otherwise. C_{Ri} is the estimated cost of executing the i^{TH} test. This cost may be in terms of power, dollars, etc. Finally, t_i is the execution time for the i^{TH} test. With this cost function as our objective, we then can use a modified version of the Time Efficient Sequence of Tests (TEST) [5,6] to construct an optimal decision tree. Its optimality will be in terms of implementation cost, acquisition cost, and resolution time as specified and weighted in Equation (5-1). The output of this phase is an optional decision tree that utilizes our optional test subset ξ_S.

5.3.3 PHASE 3: TEST PRIORITIZATION

The last phase in the BIT prioritization process involves the scoring of each of the tests in ξ_S , say N in number, based upon some BIT performance criteria. In the event that we would be unable to implement all of the members of ξ_S , we could select a subset ξ_{SS} with M elements where $M \leq N$, such that the best M tests are incorporated.

The criterion that we use is similar to that reported by [8]. It is a measure of critical information per cost returned by the test S_{Ti}.

$$S_{Ti} = \sum \omega_i / \varsigma_i \tag{5-2}$$

where S_{Ti} is the priority score for the i^{TH} test, ω_i is the weight factor the i^{TH} test, and ς_i is a cost function for the i^{TH} test.

$$\varsigma_i = A_I C_{Ii} + A_R C_{Ri} + A_T t_i \tag{5-3}$$

The terms in Equation (5-3) are the same as in Equation (5-1). The summation in Equation (5-2) is over the set of tests in ξ_S . In turn, ω_i is

$$\omega_i = \sum \alpha_{ij} P_j w_j \tag{5-4}$$

where α_{ij} , is 1 if the i^{TH} test is sensitive to the occurrence of the j^{TH} aspect, and the state of that aspect is unknown at the point of occurrence of the i^{TH} test in the decision sequence. P_j is the probability of occurrence of the j^{TH} aspect, and w_j is the importance of its occurrence. The summation occurs over the total number of aspects. In general

$$w_j = A_{w1} \zeta_{Rj} + A_{w2} \zeta_{Uj} \tag{5-5}$$

where A_{w1}, and A_{w2} are user defined coefficients, ζ_{Rj} is a criticality weight (e.g., as defined in MIL-STD-1629, Task 102), and ζ_{Uj} is a user defined measure of importance.

5.4 Summary.

A bottom-up approach for prioritizing tests for application is BIT has been defined. This approach is graphically depicted in Figure 5.3-1. The method involves three phases and yields a ranked set of tests and associated decision tree optimized for the BIT mission characteristics. Those characteristics may include fault detection, isolation, or elements of both. The method employs established techniques for construction of optimal decision trees and is highly flexible to varying design requirements. The major data requirements are knowledge of the system design, knowledge of test design and associated costs, criticality analysis results, reliability analysis results, and relative importance weights for implementation costs, execution costs, and execution times.

6.0 MIL-STD IMPACT ANALYSIS

The Automated Testability Decision Tool (ATDT) consists of a number of algorithms that address the problem of testability allocation. Specifically, those algorithms solve the following problems:

- What metrics do we use to specify testability, and how do we compute them, both during design and after deployment?
- Given a weapon system with certain testability requirements, described using the above metrics, how can we optimally allocate testability requirements for constituent subsystems, and subsequent levels of indenture?
- How do we combine our computed/measured testability metrics from lower levels of system indenture to higher levels of indenture to verify that we have met the requirements?
- How do we optimally allocate BIT as a resource?
- How do we optimally modify a design to include BIT (bottom-up)?

As shown in Figure 6.0-1, the ATDT algorithms require certain types of data. Ideally, all of them would be specified and described in the military specifications, handbooks, and standards. Similarly, the information resulting from the application of the algorithms should be described in these military documents. To the extent that a military standard, handbook, or specification describes (or should describe) the methods for generating data required for the ATDT algorithms and the formats of those data, we will consider it as a *data source*. Similarly, to the extent that such a document describes (or should describe) the format of output data resulting from the ATDT, we will regard that military standard, handbook, or specification as a *data sink*. The objective of the MIL-STD Impact Analysis was to identify those military standards, handbooks, and specifications that are either data sources or data sinks for the algorithms that comprise the ATDT. Where deficiencies were found to exist they were so identified, and are reported in this section.

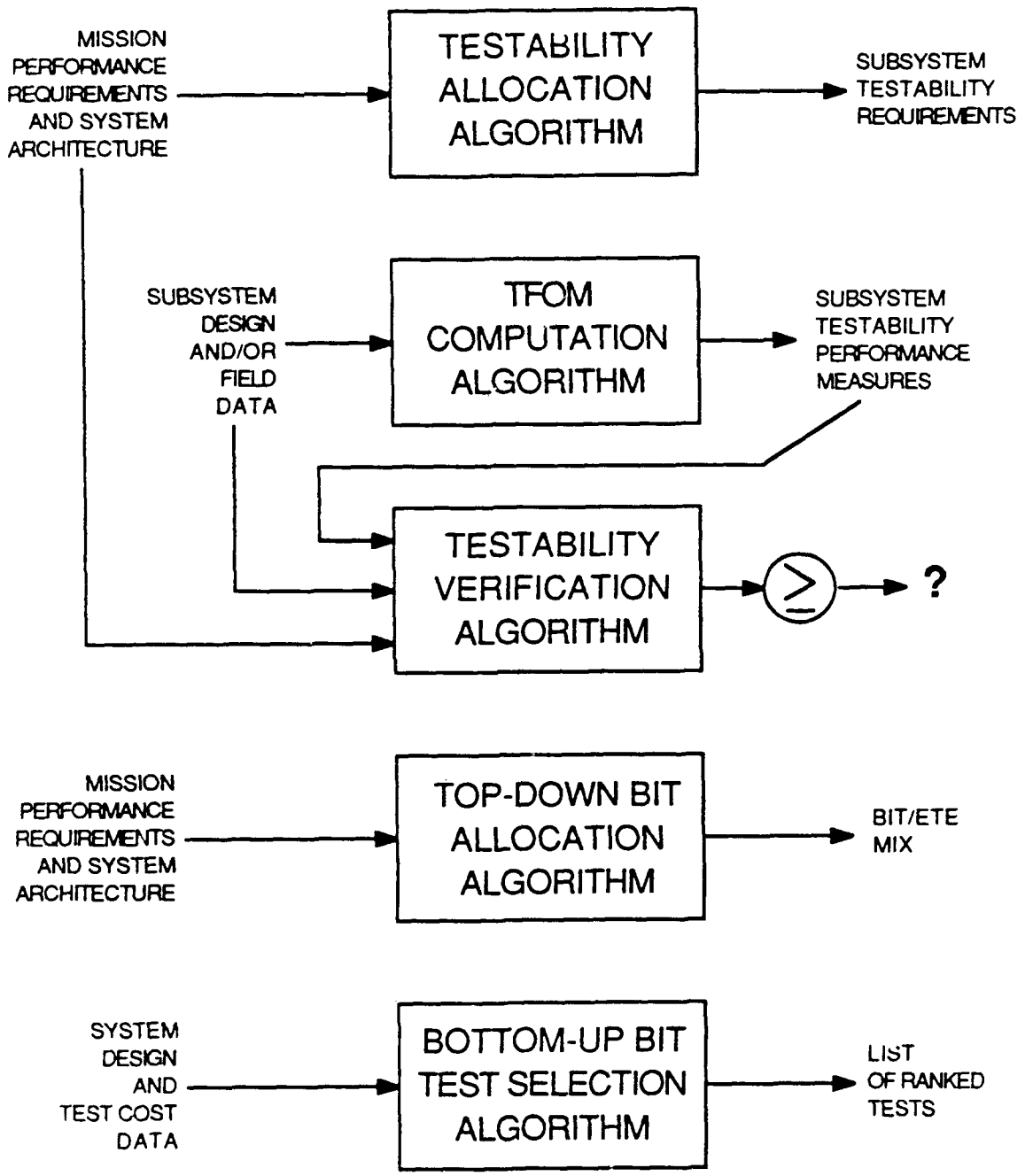


Figure 6.0-1. ATDT Algorithms and their associated input and output data.

The remainder of Section 6 is organized in three subsections. First, in Section 6.1, the data required by and output from the ATDT algorithms are collected and categorized. Then, in Section 6.2, the various data categories are used to identify those military documents that are the sources and sinks for that data. Specific recommendations are made regarding the potential for modification of the military standards in question. Finally, a summary of the impacts on those standards are given in Section 6.3.

6.1 ATDT Data Requirements.

As shown in Figure 6.0-1 and discussed in previous sections of this report, a wide variety of data is necessary to use the ATDT. Similarly, various data are generated by those algorithms. We may examine these data types for each of the five algorithms.

6.1.1 Data Types for the TAM Algorithms.

The Testability Allocation Methodology (TAM) requires the ATDT testability figures of merit as inputs and generates outputs that are also in the form of those figures of merit. In addition to TFOM's, the TAM process requires data in the form of constraints and objectives for life cycle costs, availability, operational readiness, etc. Analytical functions that relate the testability figures of merit to those constraints and objectives must also be provided. Finally, gross architectural information for the system under design must be provided along with estimated failure rates.

6.1.2 Data Types for the TFOM Algorithms.

Obviously, the TFOM Algorithms provide results in the format of the ATDT TFOM's. These algorithms require that the components comprising the system under evaluation and their failure modes be identified. Additional data required are: component failure rate estimates and relative likelihoods of their various failure modes, LSA data such as test times and costs, and design information such as signal flow and component interconnections. Finally, isolation objectives must be provided in the form of the relative importance of cost versus time.

6.1.3 Data Types for the TFOM Verification Algorithm.
The process for verification that allocated TFOM objectives have been met by a given design, requires data sources and sinks that are subsets of those for the TAM and TFOM computation algorithms. Specifically, the data sources are the system gross architectural description, estimated failure rates, and measured and required TFOM's. The result of the algorithm is a "yes/no" decision regarding the adequacy of the testability for the design in question.

6.1.4 Data Types for the Top-Down BIT Allocation Algorithm.
The algorithm for the allocation of BIT as a resource has the same data source requirements as the TAM process. The result generated by this algorithm is the relative mix of BIT versus external test methods.

6.1.5 Data Types for the Bottom-Up BIT Test Selection Algorithm.
The process for selecting tests in a design has essentially the same data source requirements as were necessary for computing TFOM's (Paragraph 6.1.2). Additional data sources required are test design and implementation costs, as well as failure mode criticality information. The output of this algorithm is a ranked list of tests that should be included in BIT.

6.1.6 Summary of ATDT Sources and Sink Data Types.
The data source types necessary for the ATDT are:

1. ATDT TFOM's
2. Performance objectives, constraints, and their associated analytical functions.
3. System architectural and design information.
4. Failure information including estimated failure rates, failure modes, and failure mode relative likelihoods.
5. LSA Test Information including execution costs and times.
6. Test implementation costs.
7. Isolation objectives -- times vs. cost priorities.

The ATDT output data types include:

1. ATDT TFOM's
2. Testability adequacy decision (Yes or No).
3. BIT vs. external test method mixing strategy.
4. Ranked list of tests for inclusion in a BIT Design.

In addition to the above, the description of all of the ATDT algorithms and their usage may be thought of as data source requirements.

6.2 MIL-STD Impact.

Some of the data source and sink requirements identified in Section 6.1.6 have little or no impact on existing military standards, handbooks or specifications. This was due to one of two situations that occurred. First, some of the data was clearly outside the scope of the military documentaion system. One example is in the case of the objective and cost functions that are TAM data source requirements. Those functions may be highly specific to a given design as a consequence of exotic technologies or unusual operational scenarios. It would be impossible for the MIL-STD's to attempt to cover all such situations. Similarly, it would be illogical to restrict the ATDT to use only such functions that may be documented in the MIL-STD's.

The second situation that occurred where the ATDT data requirements had no impact on MIL-STD's was in the case of those sources and/or sinks that are already documented in the military standards, handbooks, or specifications. An example of such are the failure-rate estimations provided in MIL-HDBK-217E.

The ATDT data types that do conflict with current military standards are:

1. The ATDT Algorithms themselves.
2. The ATDT TFOM's.
3. Failure mode relative likelihood estimations.

In the sections that follow, each of the various military documents that has been impacted by one of the above data types is reviewed. General recommendations for its potential change are called out.

6.2.1 MIL-HDBK-217E.

TITLE: Reliability Prediction of Electronic Equipment

SOURCE: Preparing Activity; RADC

PURPOSE: To establish methods for predicting the reliability of military electronic equipment and systems.

CONTENTS: The handbook contains two methods of reliability prediction; "Parts Stress Analysis", and "Parts Count". The former requires the greater amount of detail information and is applicable during the later design phase. The Parts Count Method is applicable in the early design phase and/or proposal formulation. It consists of a summation of individual failure rates with an accompanying table of these rates for convenience.

IMPACT: ATDT has a requirement for estimates of the relative likelihoods of part failure modes. Section 5 of MIL-HDBK-217E deals with prediction of component failure rates. A new section could potentially be added that apportions predicted failure rates to the individual failure modes of the parts. Data necessary to support this section has been accumulating in recent years.

6.2.2 MIL-STD-471A.

TITLE: Maintainability Verification/Demonstration/Evaluation

SOURCE: Preparing Activity; RADC

PURPOSE: To establish uniform procedures, test methods, and requirements for verifying demonstrating, and evaluating specified maintainability requirements, and for the assessment of the impact of planned logistic support. The standard is intended for use when verification, demo, and evaluation of maintainability criteria are required. The original version of MIL-STD-2165, dated 25 Jan 85, refers to this spec for demonstrating Testability criteria (Task 301).

CONTENTS: 1) Definitions & Requirements
2) Appendix A: Maintenance Task Sampling For Use With Failure Simulation
3) Appendix B: Test Methods & Failure Analysis

IMPACT: The ATDT TFOM computation and verification algorithms could be included here as a means of testability demonstration during system design. In particular, the necessary data collection to support the TFOM computations and TFOM Verifications, along with the processes themselves could be integrated in Section 4.1.1 of MIL-STD-471A, Phase I: "Maintenance Verification."

6.2.3 MIL-HDBK-472.

TITLE: Maintainability Prediction.

SOURCE: Preparing Activity; NAVAL Air Engineering Center

PURPOSE: To predict maintainability parameters of avionics, ground and shipboard electronics at the organizational, intermediate, and depot levels of maintenance.

CONTENTS: Procedure V with appendices A,B,C is appropriately found at the beginning of the handbook. This procedure is an aid for determining MTTR, fault detection and fault isolation downtimes etc. for a) early predictions and, b) detailed predictions.

IMPACT: Parameters of measure relating to isolability are potentially in conflict with the ATDT TFOM's. Specifically, "Percent Isolation to a Single Replaceable Unit" and "Percent Isolation to a Group of Replaceable Units", together possess the same information which the ATDT TFOM Fractional Isolability (FI) represents. The ATDT TFOM could replace the two (sections 3.2.2 and 3.2.3 in MIL-HDBK-472) or be added (Potentially Section 3.2.4). MIL-HDBK-472 also provides means of predicting the mean time to repair, MTTR (Section 3.2.1). This formula could be augmented by incorporating the ATDT Testability figure of merit computation for mean time to isolation.

6.2.4 MIL-STD-1591.

TITLE: On-Aircraft, Fault Diagnosis, Sub-Systems, Analysis/Synthesis of

SOURCE: Preparing Activity; RADC

PURPOSE: To establish uniform criteria for conducting trade studies to determine the optimal design for an on-aircarft fault diagnosis/isolation system.

CONTENTS: Step by step procedures for developing an on-board BIT design model based upon FFD, FFI, FFA, MTTR, man-maintenance hours, failure criticality, and probability, all balanced against cost. This standard is used for determining the costs associated with on-board testing.

IMPACT: This standard would be strongly impacted by the ATDT TFOM's. Specifically, FFI is in conflict with the ATDT TFOM Fractional Isolability (FI). In addition, no cost relationships are given that account for false alarms (i.e. FFA).

6.2.5 MIL-STD-1629-1A.

TITLE: Procedures for Performing a Failure Mode Effects and Criticality Analysis.

SOURCE: Preparing Activity; NAVAL Air Engineering Center

PURPOSE: To establish requirements and procedures for performing a Failure Mode, Effects, and Criticality Analysis (FMECA) to systematically evaluate and document, by item failure mode analysis, the potential impact of each functional or hardware failure on mission success, safety, performance, maintainability, and maintenance requirements. Each potential failure is ranked by the severity of its effect.

CONTENTS: 1) List of definitions
2) General requirements including the FMEA process
3) Task 101, FMECA
4) Task 102, Criticality Analysis
5) Task 103, FMECA-Maintainability Information
6) Task 104, Damage Mode and Effects Analysis
7) Task 105, FMECA Plan
8) Appendix A, Tasks, Rationale, and Calculations

IMPACT: Although this document would not be directly impacted by the ATDT, it could be indirectly affected. The need to determine relative failure mode likelihoods (Section 6.2.1) of this report could require that Task 101 be completed, at least in part, prior to that computation.

6.2.6 MIL-STD-2165.

TITLE: Testability Program for Electronic Systems and Equipments.

SOURCE: Preparing Activity; Naval Sea Systems Command

PURPOSE: To provide uniform procedures and methods for establishing a testability program, for assessing testability in designs, and for integration of testability into the acquisition process for electronic systems.

CONTENTS: MIL-STD-2165 is a comprehensive and explicit document that explains the Testability Program Planning Process. It is organized as follows:

1) General Requirements
2) Task 101, Testability Program Planning
3) Task 102, Testability Reviews
4) Task 103, Data Collection & Analysis Planning
5) Task 201, Requirements
6) Task 202, Preliminary Design & Analysis
7) Task 203, Detail Design & Analysis

8) Task 301, Inputs to Maintainability Demo
9) Appendix A, Testability Program Application Guidance
10) Appendix B, Inherent Testability Checklist
11) Appendix C, Glossary
12) Testability Flow Charts

IMPACT: There are numerous areas within MIL-STD 2165 that could be afected by the ATDT. Specifically, the TAM and BIT Allocation Algorithms could be cited by, and/ or integrated into, Task 201. Similarly, Tasks 202 and 203 could incorporate other ATDT algorithms. In particular, Task 202.2.2 could cite the ATDT BIT Allocation algorithm; Task 203.2.8 could cite the ATDT Bottom-Up BIT prioritization algorithm; and Tasks 202.2.3 and 203.2.3 could both incorporate or cite ATDT algorithms for TFOM computation and testability verification.

6.2.7 AFSC DH 1-9.

TITLE: AFSC Design Handbook 1-9, Maintainability

SOURCE: Preparing Activity; Air Force Systems Command

PURPOSE: To provide system designers with maintainability design principles for ground electronics, nondestructive inspection (NDI), and on-condition maintenance.

CONTENTS: This handbook deals primarily with maintainability figures of merit, design, allocation, costs, and their interrelationships. Chapter 3 details an allocation process for an actual system. Chapter 4 examines in some detail some of the problems and alternatives in BIT design. Its organization is as follows:

Chapter 1	General Information
Chapter 2	Establish Maintainability and Maintenance Requirements
Chapter 3	Maintainability Design Processes
Chapter 4	Factors that Influence Maintainability
Chapter 5	Design Controls
Chapter 6	Maintainability Tests & Demonstration
Chapter 7	Mathematical and Statistical Concepts

Chapter 8 NDI and On-Condition Maintenance
Appendix A Glossary of Engineering Terms
Appendix B DOD Index of Specs and Standards
Index

IMPACT: In Chapter 2, Design Note 2B1, "Trade-Off Processes" could be supplemented or another Design Note added that discusses the TAM algorithm. Design Note 3B2, "Basic Steps of Allocation" could also benefit from the incorporation of the TAM algorithm.

6.2.8 GIMADS: MIL-STD-xxxx.

TITLE: Generic Integrated Maintenance Diagnostics

SOURCE: USAF, Aeronautical Systems Division

PURPOSE: This standard is designed to be used by both the Air Force and its contractors to establish requirements for incorporating the programmatic aspects of integrated diagnostics into weapon system procurements.

CONTENTS: The GIMADS standard includes requirements and verification for a generic integrated diagnostics process that can be tailored for application to a specific weapon system. Appendix H, a roadmap, is included to facilitate selecting requirements. Appendix A contains rationale, guidance, and lessons learned for tailoring and implementing requirements and verifications.

IMPACT: There are various tasks called out in the GIMADS standard that could be impacted by the processes contained in the ATDT. The process of testability allocation, GIMADS task 4.1.3.4.1 could cite the ATDT TAM algorithm as a means for allocation to the lower levels of system indenture. Both the TAM and BIT Allocation algorithms in the ATDT could be applied in the preliminary design task, 4.1.4.4.1. Further, within the preliminary design, Subtask 4.1.4.4.1.6 calls for an inherent testability assessment. Both the ATDT TFOM computation and testability verification algorithms could have roles here. Task 4.1.4.4.5, the Diagnostic Detailed Design, could also employ these two ATDT algorithms.

6.3 MIL-STD Impact Summary and Conclusions.

There are various MIL-STD's that could be impacted by the ATDT. In some cases, that impact is minor (e.g. situations where FFI is cited as opposed to an ATDT TFOM). However, the impact on other standards could be substantial. The documents that are most impacted are MIL-HDBK-217E, MIL-STD-2165, and GIMADS.

MIL-HDBK-217E could be affected as a result of the need, on the part of the ATDT TFOM computation algorithm, to extend the failure rate analysis to the level of piece part failure modes. This could require an addition of an entire new section.

MIL-STD-2165 and GIMADS are strongly impacted by the ATDT as a consequence of their scope. Both of those standards encompass most aspects of the diagnostic/testability requirements, design, and development process. The ATDT algorithms are designed to be used for allocation prior to design, and verification during and after design. It is therefore entirely expected that MIL-STD-2165 and GIMADS would be affected by those algorithms.

7.0 BIBLIOGRAPHY AND REFERENCES

7.1 TFOM Bibliography

The following bibliography lists all the pertinent reference material that was used in the TFOM study.

Abramovici, M and Breuer, M. A. *"Fault Diagnosis in Synchronous Sequential Circuits Based on an Effect-Cause Analysis ,"* IEEE Transactions on Computers; Vol. C-31, No. 12, December 1982; PP 1165-1172.

Abramovici, M.,*"Multiple Fault Diagnosis in Combinational Circuits Based on an Effect-Cause Analysis ,"* IEEE Transactions on Computers; Vol. C-29, No., 6, June 1980; PP. 451-460.

Aly, A. A., Elsayedaly, N. A., *"An Efficient Algorithm for Optimal Design of Diagnostics ,"* IEEE Transactions on Reliability: Vol. R-32, No. 5, December 1983; PP426-432.

Bearzi, B. , Fenoglio, F., Turconi G., *"Diagnostic Coverage as Life Cycle Cost Parameter "*, Reliability in Electrical and Electronic Components and Systems, E. Lauger and J. Moltoft (Editors), North-Holland Publishing Company, 1982.

Bhavsar D. K. *"Design for Test Calculus: An Algorithm for DFT Rules Checking Proceedings of the 20TH Design Automation Conference "*, Proceedings of the 20TH Design Automation Conference; PP. 300-307.

Bossen, D. C. , Hsiao, M. Y., *"Model for Transient and Permanent Error Detection and Fault Isolation Coverage "*, IBM Journal of Research and Development; Vol. 26, No. 1, January 1982.

Bussert, J. , *"Testability Analysis Tools on a Military System "*, Technical Report TM-3143-1717; Navy Test Technology; Naval Ocean Systems Center; Naval Weapons Station, Seal Beach, Corona Annex; Fleet Analysis Center, Corona, CA 91720-5000; September 1987.

Chang, H.Y. , "*An Algorithm for Selecting an Optimum Set of Diagnostic Tests* ",
IEEE Transactions on Electric Computers; Vol. EC-14, No. 5, October 1965; PP 706-711.

Chen, H. S. M., Saeks, R. "*A Search Algorithm for the Solution of Multifrequency Fault Diagnosis Equations* ", IEEE Transactions on Circuits and Systems; Vol. CAS-26, No. 7, July 1979; PP 589-594.

Cohn, M., Ott, G., "*Design of Adaptive Procedures for Fault Detection and Isolation* ", IEEE Transactions on Reliability; Vol. R-20, No. 1, February 1971, PP. 7-10.

Committee on Isolation of Faults in Air Force Weapons and Support Systems Air Force Studies Board. Commision on Engineering and Technical Systems, National Research Council, "*Isolation cf Faults in Air Force Weapons and Support Systems, Volume 1*" , National Academy Press, Washington DC, 1986.

Cook, T. N., Ariano, J., "*Analysis of Fault Isolation Criteria/Techniques*" , 1982 Proceedings Annual Reliability and Maintainability Symposium, Los Angeles, CA, 1982, P. 206.

Dahbura, A. T., Masson, G. M., "*A New Diagnosis Theory as the Basis of Intermittent-Fault/Transient-Upset Tolerant System Design*" .

Danner, F. G. "*System Test Visibility -- Or Why Can't You Test Your Electronics?*", Proceedings 1983 IEEE International Test Conference, PP. 635-639.

Brendan, D. , "*The Economics of Automatic Testing* ", McGraw-Hill Book Company, LTD., London, England, 1982.

Freeman, S. , "*Optimum Fault Isolation by Statistical Inference*", IEEE Transactions on Circuits and Systems; Vol. CAS-26, No 7, July 1979; PP. 505-512.

Gardner, W.A. , "*Likelihood Sensitivity and the Cramer-Rao Bound*", IEEE Transactions on Information Theory; Vol. IT-25, No., July 1979, P491

Genesereth, M.R., "*Diagnosis Using Hierarchical Design Models*" Proceedings AAAI-82, Pittsburgh, PA, August 1982, PP. 278-283.

Gilreath, A. E., Kelley, B. A., Simpson, W. R., "*Organizational-Testability Attributes*", Final Technical Report, Rome Air Development Center, Grifiss Air Force Base, NY 1 3441-5700, November 1986.

Goel, P., "*Test Generation Costs Analysis and Projections*", Proceedings of the 17th Design Automation Conference, Minneapolis, MN, June 1980, PP. 77-84.

Grason, J., Nagle, A. W., "*Digital Test Generation and Design for Testability*", Proceedings of the 17th Design Automation Conference, Minneapolis, MN, June 1980, PP. 175-189.

Greenspan, A. M., "*Establishing Testability Standards*", Proceedings Autotestcon '78, San Diego, CA, November, 1978, PP. 275-281.

Greenspan, A. M., Myles, M. D. "*Perspectives on Testability*", Proceedings Automatic Testing 79, Conference Proceedings, Part I, Brighton, England, December 1979, PP. 1-9.

Hartmann, C. R. P., Varshney, P. K., Mehrotra, K. G., Herberich, C. L., "*Application of Information Theory to the Construction of Efficient Decision Trees*", IEEE Transactions on Information Theory; Vol IT-28, No. 4, July 1982, PP. 565-577.

Holbrook, R. O., "*BIT Detectability/Reliability in Expendable Weapons*", 1984 Proceedings Annual Reliability and Maintainability Symposium; PP. 306-311.

Johnson, A. T. Jr., *"Efficient Fault Analysis in Linear Analog Circuits "*, IEEE Transactions on Circuits and Systems; Vol. CAS-26, No. 7, July 1979; PP. 475-484.

Johnson, R. A. *"An Information Theory Approach to Diagnosis "*, Proceedings of the 6th Annual Conference on Reliability and Quality Control, PP. 102-109, January 1960.

Lahore, H., *"Artificial Intelligence Applications to Testability "*, Final Technical Report RADC-TR-84-203, Rome Air Development Center, Air Force Systems Command, Griffiss Air Force Base, NY 13441, (Boeing Aerospace Company), October 1984.

Lederer, P. S., *"Sensor Handbook for Automatic Test, Monitoring, Diagnostic, and Control Systems Applications to Military Vehicles and Machinery "*, PB82-123746, Center for Electronics and Electrical Engineering, National Engineering Laboratory, National Bureau of Standards, Washington, DC 20234, October 1981.

Lee, J., Bedrosian, S. D., *"Fault Isolation Algorithm for Analog Circuits Using the Fuzzy Concept "*, IEEE Transactions on Circuits and Systems; Vol. CAS-26, No. 7, July 1979; PP 518-522.

Lee, R. E., *"Logistical Impacts Within the Cost Analysis Community "*, Armed Forces Comptroller; Spring 1983; PP. 18-20.

Locurto, C. A., *"Impact of BIT on Avionics Maintainability "*, 1983 Proceedings Annual Reliability and Maintainability Symposium, PP. 333-338.

Malcolm, J. G., *"The Need: Improved Diagnostics-Rather Than Improved R "*, 1984 Proceedings Annual Reliability and Maintainability Symposium, PP. 315-322.

Malcolm, J. G. *"Practical Application of Bayes' Formulas ,"* 1983 Proceedings Annual Reliability and Maintainability Symposium, PP. 180-186.

Malcolm, J. G., *"BIT False Alarms: An Important Factor in Operational Readiness ,"* 1982 Proceedings Annual Reliability and Maintainability Symposium, PP. 206-211.

Mehra, R. K., *"Optimal Input Signals for Parameter Estimation in Dynamic Systems -- Survey and New Results ,"* IEEE Transactions on Automatic Control; Vol. AC-19, No. 6, December 1974, PP. 753-768.

Motohara, A., Fujiwara, H., *"Design for Testability for Complete Test Coverage ,"* IEEE Design & Test, November 1984; PP. 25-32.

Muehldorf, E. I., Savkar, A. D., *"LSI Logic Testing -- An Overview ,"* IEEE Transactions on Computers; Vol. C-30, Nol 1, January 1981; PP. 1-16.

Navid, N., Willson, A. N., Jr., *"A Theory and an Algorithm for Analog Circuit Fault Diagnosis ,"* IEEE Transactions on Circuits and Systems, Vol. CAS-26, No. 7, July 1979; PP. 440-457.

Neumann, G., *"Built in Effectiveness Study ,"* Draft Report USN 1124/0679/BD14-2, Fleet Analysis Center, NWS Seal Beach, Corona Annex, (Giordano Associates, Inc.).

Ozawa, T. and Kajitani, Y., *"Diagnosability of Linear Active Networks ,"* IEEE Transactions on Circuits and Systems; Vol. CAS-26, No. 7, July 1979, PP. 485-489.

Pattipati, K. R. , Alexandridis, M. G., and Deckert, J. C., *"A Heuristic Search/Information Theory Approach to Near-Optimal Diagnostic Test Sequencing ,"* Proceedings of the 1986 IEEE International Conference on Systems, Man, and Cyberactics, Vol. 1, Atlanta, GA, October, 1986, PP. 230-255.

Pattipati, K. R., Deckert, J. C., *"Computer-Aided Design Techniques for Automated Test Program Development: Phase I Final Report ,"* Final Report (Grant No. ECS-8460598), National Science Foundation, SBIR/Room 1250, 1800 G. Street, NW, Washington, DC 20550, (Alphatech, Inc.), July 1985.

Pattipati, K. R., Willsky, A. S., Deckert, J. C., Eterno, J. S., Weiss, J. S., "*A Design Methodology for Robust Failure Detection and Isolation* ," Proceedings of the 1984 American Control Conference, Vol. 3, San Diego, CA, June 1984, PP. 1755-1762.

Peterson, J. L. , "*Petri Net Theory and the Modeling of Systems* ," 1981 Prentice-Hall, Inc., Englewood Cliffs, NJ 07632.

Pliska, T. F., Jew, F. L., Angus, J. E. ,"*BIT/External Test Figures of Merit and Demonstration Techniques* ," Final Technical Report RADC-TR-79-309, Rome Air Development Center, Air Force Systems Command, Griffiss Air Force Base, NY 13441 (Hughes Aircraft Company), December 1979.

Priester, R. W., Clary, J. B., "*New Measures of Testability and Test Complexity for Linear Analog Failure Analysis* ," IEEE Transactions on Circuits and Systems; Vol. CAS-28, No. 11, November 1981, PP. 1088-1092.

Rosenberg, B. J., "*Integrated Diagnostic Support System (IDSS) Weapon System Testability Analyzer Program Design Specification ,"* (Contract No. N00024-87-C-4033) Department of the Navy, Naval Sea Systems Command, Washington, DC 20362, (Harris GSSD), December 1986.

Roth, J. P., "*Diagnosis of Automata Failures: A Calculus and a Method* ," IBM Journal of Research and Development, Vol. 10, No. 4, 1966, PP. 278-291.

Sen, N., Saeks, R., "*Fault Diagnosis for Linear Systems Via Multifrequency Measurements* ," IEEE Transactions on Circuits and Systems; Vol. CAS-26, No. 7, July 1979; PP. 457-465.

Simpson, W. R., *"Organizational Maintenance: A Modified State Representation* ," Proceedings Autotestcon 85;, Uniondale, N.Y. October 1988, PP. 400-405.

Simpson, W. R., *"Active Testability Analysis and Interactive Fault Isolation using Stamp ,"* Proceedings Autotestcon 87, San Francisco, CA, November 1987, PP. 105-111.

Simpson, W. R., "*Stamp Testability and Fault-Isolation Applications .*"

Simpson, W. R., Dowling, C. S., *"Wraple: The Weighted Repair Assistance Program Learning Extension ,"* IEEE Design & Test, April 1986; PP. 66-73.

Sridhar, T., Hayes, J. P., "*Design of Easily Testable Bit-Sliced Systems ,"* IEEE Transactions on Circuits and Systems; Vol. CAS-28, No. 11, November 1981; PP. 1046-1058.

Stenbakken, G. N., Souders, M. T., "*Test-Point Selection and Testability Measures Via Q-R Factorization of Linear Models ,"* IEEE Transactions on Instrumentation and Measurement; Vol. IM-36, No. 2, June 1987; PP. 406-410.

Swets, J. A., Pickett, R. M., "*Evaluation of Diagnostic Systems ,"* Academic Press, Inc., New York, NY 10003, 1982.

Varshney, P. K. , Hartmann, C. R. P. & De Faria, Jr., J. M., *"Application of Information Theory to Sequential Fault Diagnosis ,"* IEEE Transactions on Computers; Vol. C-31, No. 2, February 1982; PP. 164-170.

Visvanathan, V. & Vincentelli, A. S., "*Diagnosability of Nonlinear Circuits and Systems -- Part II: Dynamical Systems ,"* IEEE Transactions on Circuits and Systems; Vol. CAS-28, No. 11, November 1981; PP 1103-1108.

Visvanathan, V. & Vincentelli, A. S.,*"Diagnosability of Nonlinear Circuits and Systems -- Part I: The DC Case ,"* IEEE Transactions on Circuits and Systems; Vol. CAS-28, No. 11, November 1981; PP 1093, 1102.

Willsky, A. S., "*A Survey of Design Methods for Failure Detection in Dynamic Systems ,"* Automatica, Vol. 12, Pergamon Press, 1976; PP 601-611.

"Built-In-Test Equipment Requirements Workshop ," by Workshop Presentation; Paper P-1600, Institute for Defense Analyses, Program Analysis Division, 400 Army-Navy Drive, Arlington, VA 22202, August 1981.

Zaghioul, M. E., *"Testability Measures for the design of Digital IC's ,"* Semicustom Design Guide 1987; PP. 98-108.

7.2 TAM and TFOM/TAM References and Bibliography

The following lists all the pertinent reference material that was used in the TAM and TFOM/TAM study.

[1] Charnes, A. and Cooper, W., "*The Theory of Search: Optimum Distribution of Search Effort*," Management Sci., Vol 5 (1958), pp. 44-49.

[2] Luss, H. and Gupta, S. K., "*Allocation of Effort Resources Among Competing Activities*," Operations Research, Vol 23 (1975), pp. 360-366.

[3] Wilkinson, C. and Gupta, S. K., " *Allocating Promotional Effort to Competing Activities: A Dynamic Programing Approach*," IFORS Conference, Venice, 1969, pp. 419-432.

[4] Bodin, L., "*Optimization Procedures for the Analysis of Coherent Structures*," IEEE Trans. Reliab., Vol. R-18 (1969), pp. 118-126.

[5] Bitran, G. and Hax, A., "*Disaggregation and Resource Allocation Using Convex Knapsack Problems With Bounded Variables*," Management Sci., Vol 27, (1981), pp. 431-441.

[6] Held, M., Wolfe, P. and Crowder, H., "*Validation of Subgradient Optimization," Math. Programming*, Vol 6 (1974), pp. 68-88.

[7] Zipkin, P. H., "*Simple Ranking Methods For Allocation of One Resource*," Management Sci., Vol 26 (1980), pp. 34-43.

[8] Everett, H., "*Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources*," Operations Res., Vol 11 (1963), pp. 399-417.

[9] Karush, W., "*A General Algorithm for the Optimal Distribution of Effort*," Management Sci., Vol 9 (1962), pp. 50- 72.

[10] Koopman, B., "*The Theory of Search: III. The Optimum Distribution of Searching Effort*," Operations Res., Vol 5 (1957), pp. 613-629.

[11] De Gueni, J., "*Optimal Distribution of Search Effort*," Operations Res., Vol 9 (1961), pp. 1-7.

[12] Greenberg, H. and Pierskalla, W., "*Surrogate Mathematical Programming*," Operations Res., Vol 18 (1970), pp 924-939.

[13] Shih, W., "*A New Application of Incremental Analysis in Resource Allocations*," Operational Res. Quart. 25 (1974), pp. 587-597.

[14] Mjelde, K. M., "*The Optimality of an Incremental Solution of a Problem Related to the Distribution of Effort*," Operational Res. Quart. 26 (1975), pp. 867-870.

[15] Einbu, J. M, "*On Shih's Incremental Method in Resource Allocations*," Operational Res. Quart. 28 (1977), pp. 459-462.

[16] Danskin, F. M, "*The Theory of Max-Min*," Springer-Verlag, 1967, pp. 85-100.

[17] Mjelde, K. M., "*Evaluation and Incremental Determination of Almost Optimal Allocations of Resources*," Operational Res. Quart. 27 (1976), pp. 581-588.

[18] Einbu, J. M., "*Extension of the Luss-Gupta Resource Allocation Algorithm by Means of First Order Approximation Techniques*," Operations Res., Vol 29 (1981), pp 621-626.

[19] Geoffrion, A., "*Elements of Large-Scale Mathematical Programming*," Management Sci., Vol 16 (1970), pp. 652-691.

[20] Bertsekas, D. P. " *Multiplier Methods: A Survey,*" Automatica, Vol 12 (1976), pp 133-145.

[21] Allen, D., Joe, E., Fleming, R. and Josselyn, J. V., " *Testability Allocation and Program Monitoring for Fault-Tolerant Systems Prior To Detailed Design,*" Proc. 1987 AUTOTESTCON, San Francisco, California, November 1987, pp 441-446.

[22] Caroll, W. H., Linden, V. L. and Waldo, C. R., " *Diagnostic Specification-A proposed Approach,* " IEEE Trans. Reliability, Vol R-30, No.3, August 1981, pp 227-231.

[23] Malcolm, J. G., Highland, R. W., " *Analysis of Built-In-Test (BIT) False Alarm Conditions,*" RADC-TR-81, 1981 February.
↳ -220

[24] Harris, D. E. "*Built-In-Test for Fail-Safe Design,*" 1986 Proc. Annual Reliability and Maintainability Symposium, pp 361-366.

MIL-STD-1591: *On Aircraft, Fault Diagnosis, Subsystems Analysis/ Synthesis of*

MII-HDBK-472: *Maintainability Prediction*

MIL-STD-756A: *Reliability Modeling and Prediction*

MIL-HDBK-217: *Reliability Stress and Failure Rate Data for Electronic Parts*

MIL-STD-499A: *Engineering Management*

MATE GUIDE 3: *Avionics Testability Design Guide*, REV C., April 1985.

MIL-STD 2165: *Testability Program for Electronic Systems and Equipments*

Kernighan, B.W., & Lin, J., "*An Efficient Heuristic Procedure for Partitioning Graphs ,*" Bell Systems Technical Journal, 49, pp 291-307, 1970.

Balinski, M.L., "*Integer Programming: Methods, Uses and Computation*," Management Sciences, 12, 1965, pp. 253-313.

Barlow, Hunter & Proschan, "*Optimum checking Procedures*," J. Soc Indus Appl Math, 11, 1963, pp. 1078-1095.

Bellmore & Nemhauser, "*The Traveling Salesman Problem: A Survey*," Operations Res., Vol. 16, 1968, pp. 538-558.

Bertsekas, D. P., "*Constraints Optimization and Lagrange Multiplier Methods*," Academic Press, New York.

McLeavey, D.W., McLeavey, J.A., "*Parallel Optimization Methods in Standby Reliability*," University of Connecticut, School of Business Administration, Bureau of Business Research, Working Paper, No. 2, 12 pages, 1975.

Leung, F., & David, K.H., "*An Optimum Allocation of Different Weapons to a Target Complex*," Operation Res., Vol. 11, 1963, pp. 787-794.

Tillman, F.A., "*Integer Programming Solutions to Constrained Reliability Optimization Problems*," Transactions of Twentieth Annual Technical Converence, American Society for Quality Control, Paper Number 66-174, 1966, pp. 676-693.

Lawler, F.L. & Wood, D.E., "*Branch-and-Bound Methods: A Survey*," Operations Res., Vol. 14, pp. 699-719, 1966.

Moskowitz, F. & McLean, J.B. "*Some Reliability Aspects of System Design*," IRE Trans Reliability and Quality Control, Vol. PGROC-8, Sept. 1956, pp. 7-35.

Proschan, F. & Bray, T.A. "*Optimum, Redundancy Under Multiple Constraints*," Operations Res., Vol. 13, No. 5, 1965, pp. 800-814.

Dantzig, G. B., Fulkerson, D. R. & Johnson, S. M., "*On a Linear Programming Combinatorial Approach to the Traveling Salesman Problem*," Operations Res., Vol. 7, No. 1, 1959, pp. 58-66.

Hadley, G., "*Nonlinear and Dynamic Programming* ," Addison Wesley, Reading, Mass.

Garfinkel & Nemhauser, "*Integer Programming* ," Wiley, New York 1972.

Geoffrion, A.M., & Marsten, R.E., "*Integer Programming Algorithms: A Framework, and State-of-the-Art Survey* ," Management Science, Vol. 18. No. 9, 1972, pp. 465-491.

Gill, P. E. & Murray, W., "*Numerical Methods for Constrained Optimization* ," Academic Press.

Gluss, B., "*An Optimum Policy for Detecting a Fault in a Complex System* ," Operations Res., Vol. 7, 1959.

Chang, H. Y., "*An Algorithm for Selecting an Optimum Set of Diagnostic Tests* ," IEEE TRansactions on Electronic Computers, Vol. EC-14, No. 5, October 1965.

Brule, J. D., Johnson, R. A. & Kletsky,E. J., "*Diagnosis of Equipment Failures* ," IRE TRansactions on Reliability & Quality Control, Vol. RQC-9, pp. 23-24, April 1960.

Kettelle, J. D., "*Least-Cost Allocation of Reliability Investment* ," Operations Res., Vol. 10, pp. 249-265 (March-April 1967)

DeCorlieu, J., " *Maintainability Diagnosis Techniques,*" 1966 Annual Symposium on Reliability Systems.

Coleman, J. J. & Abrams, J., "*Mathematical Model for Operational Readiness* ," Operations Res., Vol 10, No. 1, 1962, pp 126-136.

Misra, K.B., "*A Method for Redundancy Allocation,*" Microelectronics and Reliability, Vol. 12, Oct. 1973, pp. 389-393.

Misra, K.B. & Carter, C.E., "*Redundancy Allocation in a System with Many Stages*," Microelectronics and Reliability, Vol. 12, June 1973, pp. 223 228.

Aggarwal, K. K., Misra, K. B. & Gupta, J. S., "*Reliability Evaluation: A Comparative Study of Different Techniques*, " Microelectronics and Reliability, Vol. 14, 1975, pp. 49-56.

Srikantan, K. S., "*A Problem in Optimum Allocation* ," Operation Res., Vol. 11, No. 2, 1963, pp. 265-273.

Wattanapanou, N. & Shaw, L., "*Optimal Inspection Schedules for Failure Detection in a Model Where Tests Hasten Failures* ," Operations Res., Vol. 27, No. 2, 1979, pp. 303-317.

Polak, E. "*Computational Methods in Optimization: A Unified Approach*," Academic Press.

Bellman R. & Dreyfus S., "*Dynamic Programming & the Reliability of Multicomponent Devices*, " Operations Res., Vol. 6, No. 2, 1958, pp. 200-206.

Bellman R. & Dreyfus S., "*Applied Dynamic Programming*," Princeton University Press, Princeton, New Jersey.

Fleming, R. E. , Josselyn, J. V. & Boyle, P., "*Integrated Supportability Analysis* ," NAECON 1987.

Dreyfus S E. & Law, A. M., "*The Art and Theory of Dynamic Programming*," Academic Press, New York 1977.

Morin, T.L. & Marsten, R., "*An Algorithm for Nonlinear Knapsack Problems*," Management Sciences, Vol. 22, No. 10, 1976, pp. 1147-1158.

Weingartner, H., Martin, G. & Ness, D. N., "*Methods for the Solution of the Multi-Dimensional 0/1 Knapsack Problem*, " Operations Research, Vol. 15, No. 1, Jan-Feb 1967, pp. 83-103.

Shershin, A.C., "*Mathematical Optimization Techniques for the Simultaneous Apportionments of Reliability and Maintainability*," Operations Research, Vol. 18, pp. 95-106, Jan-Feb 1970.

Winter, B.B., "*Optimal Diagnostic Procedures*," IRE Transactions on Reliability and Quality Control, Vol. RQC-9, No. 3, pp. 13-19, Dec 1960.

Chang, C.L., & Slagle, J.R., "*An Admissible and Optimal Algorithm for Searching AND/OR Graphs*," Artificial Intelligence, Vol. 2, 1971, pp. 117-128.

Lie, C.H., Hwang, C.L., & Tillman, F.A., "*Availability of Maintained Systems: A State of the Art Survey*," AIIE Transactions 1977.

Bertsekas, D.P., "*Projected Newton Methods for Optimization Problems with Simple Constraints*," SIAM Journal on Control & Optimization, Vol. 20, No. 2, March 1982, pp. 221-246.

French & Al, "*Multi-Objective Decision Making*," Academic Press.

Black, G., & Proschan, F. "*On Optimal Redundancy*," Operations Res., Vol. 7, 1959, pp. 581-588.

Salkin, H.M. & Dekluyer, C.A., "*The Knapsack Problem: A Survey*," Technical Memo, No. 281 Department of Operations Research, Case Western Reserve University, Revised May 1973.

Swets, J. A. & al, "*Assessment of Diagnostic Technologies*," Science, Vol. 205, No. 4408, August 1979.

Koopman, B.O., "*The Theory of Search I & II*," Operations Research, Vol. 4, 1956, pp. 324-346, & 503-531.

Krone, "*Heuristic Programming Applied to Scheduling Problem*," Proceeding of the 5th Annual Conference Information Science Systems, Dept. of Electrical Engineering, Princeton, University, 1971.

Webster, L.R., "*Optimum System Reliability and Cost Effectiveness*," Proc. 1967 Annual Symposium on Reliability, pp. 489-500.

Lawler, E.L., & Bell, M.D., "*A Method for Solving Discrete Optimization Problems*," Operations Res., Vol. 14, pp 1098-1112, 1966.

Lin, S & Kernighan, B.W., "*An Effective Heuristic Algorithm for the TSP*," Operations Res., Vol. 21, pp 498-516, 1973.

Malck, M., & Liu, K., "*Graph Theory Models in Fault Diagnosis and Fault Tolerance*," Design Automation and Fault-Tolerant Computing, Vol. III, Issue 3/4, 1980.

Raghavachari, M., "*On Connections Between Zero/One Integer Programming and Concave Programming Under Linear Constraints*," Operation Res. Vol. 17, No. 4, 1969, pp 680-684.

Martelli & Montanari, U., "*From Dynamic Programming to Search Algorithms with Functional Costs*," Proceedings of the Fourth International Joint Conference Artificial Intelligence, Tiblisi, Sept 1975, pp 345-350.

Martelli & Montanari, U., "*On the Foundations of Dynamic Programming in Topfics in Combinatorial Optimization*," S. Rinaldi(eds), Springer Verlag, 1975, pp 145-163.

Muckstadt, J. & Koenig, S.A., "*An Application of Lagrangian Relation to Scheduling in Power Generation Systems*," Operations Res., Vol. 25, pp 387-403, 1977.

Gilmore, P.C., Gomory, R.F., "*The Theory and Computation of Knapsack Functions*," Operations Research 14, pp 1045-1074, 1966.

Pattipati, Kastner & Shaw et Al, "*A Hierarchical Model for the Design of Large Multi-Shop Maintenance Facilities*," Proceedings of the IEEE Conference on Systems, Man and Cybernetics, New Delhi, India, January 1983, pp 560-568.

Furstman, S. & Gluss, B., "*Optimum Search Routines for Automatic Fault Location*," Operations Research, Vol. 8, 1960, pp 512-523.

Bokhari, S.H., "*Dual Processor Scheduling with Dynamic Reassignment*," IEEE Trans Software Engrg, SE-5, 341-349, 1979.

Zahl, S., "*An Allocation Problem with Applications to Operations Research and Statistics*," Operation Research, Vol. 11, No. 3, 1963, pp 426-441.

Sandell, Bertsekas, Shaw, Gully & Gendron, "*Optimal Scheduling of Large-Scale Hydrothermal Power Systems*," Proceedings of the 1982 IEEE International Conference on Large-Scale Systems Symposium, Virginia Beach, VA, 1982, pp 141-147.

Shapiro, J.F., "*A Survey of Lagrangian Techniques for Discrete Optimization*," Annals Discrete Math, Vol. 5, pp 113-118, 1979.

Sheskin, T.J., "*Partitioning of Modular Equipment for Fault Isolation*," Microelectronic Reliability, Vol. 17, Pergamon Press, Ltd, 1978.

Thomas, F. A., "*The Concept of Coverage and Its Effect on the Reliability Model of a Repairable System*," 1972 International Symposium on Fault-Tolerant Computing, Newton, MA, June 1972.

7.3 References for the Bottom-Up BIT Prioritization

1. Pliska, T. F., Jew, F. L., and Angus, J. E., "*BIT/External Test Figures of Merit and Demonstration Techniques*," Final Technical Report RADC-TR-79-309, Rome Air Development Center, Air Force Systems Command, Griffiss Air Force Base, NY 13441, December 1979.

2. Franco, J. R. Jr., "*Experiences Gained Using the Navy's IDSS Weapon System Testability Analyzer*," Proceedings Autotestcon '88, Minneapolis-St. Paul, Minnesota, October 1988, PP. 129-132.

3. Simpson, W. R., "*The Application of the Testability Discipline to Full Systems Analyses*," Proceedings 1983 IEEE Automatic Test Program Generation Workshop, San Francisco, California, March, 1983.

4. Navid, N. & Willson, A. N. Jr., "*A Theory and an Algorithm for Analog Circuit Fault Diagnosis*," IEEE Transactions on Circuits and Systems, Vol. CAS-26, No. 7, July 1979, PP. 440-457.

5. Pattipati, K. R., Alexandridis, M. G., & Deckert, J. C., "*A Heuristic Search/Information Theory Approach to Near-Optimal Diagnostic Test Sequencing*," Submitted to IEEE Transactions on Systems, Man, and Cybernetics, July 1988.

7. Pattipati, K. R., Private communications with author concerning test sequencing in modular systems.

8. Simpson, W. R., Private communications with author concerning test selection criteria for BIT application.

APPENDIX A

TESTABILITY BURDEN ESTIMATION DATA

This appendix provides the figures and tables necessary to compute the hardware relative overhead burden of BIT/BITE testability features for various levels of the probability of fault isolation Pr(I) in avionics equipment.
These figures and tables are reprinted from the MATE Guide G3V3P2 section 7 and appendix E. The numbering scheme used for the figures is the same as the one used in the MATE Guide. However, due to the omission of some MATE figures and in order to keep a logical and consecutive numbering order, there are exceptions which are so noted. In particular, the tables are referenced differently (i. e., T-X instead of E-X).

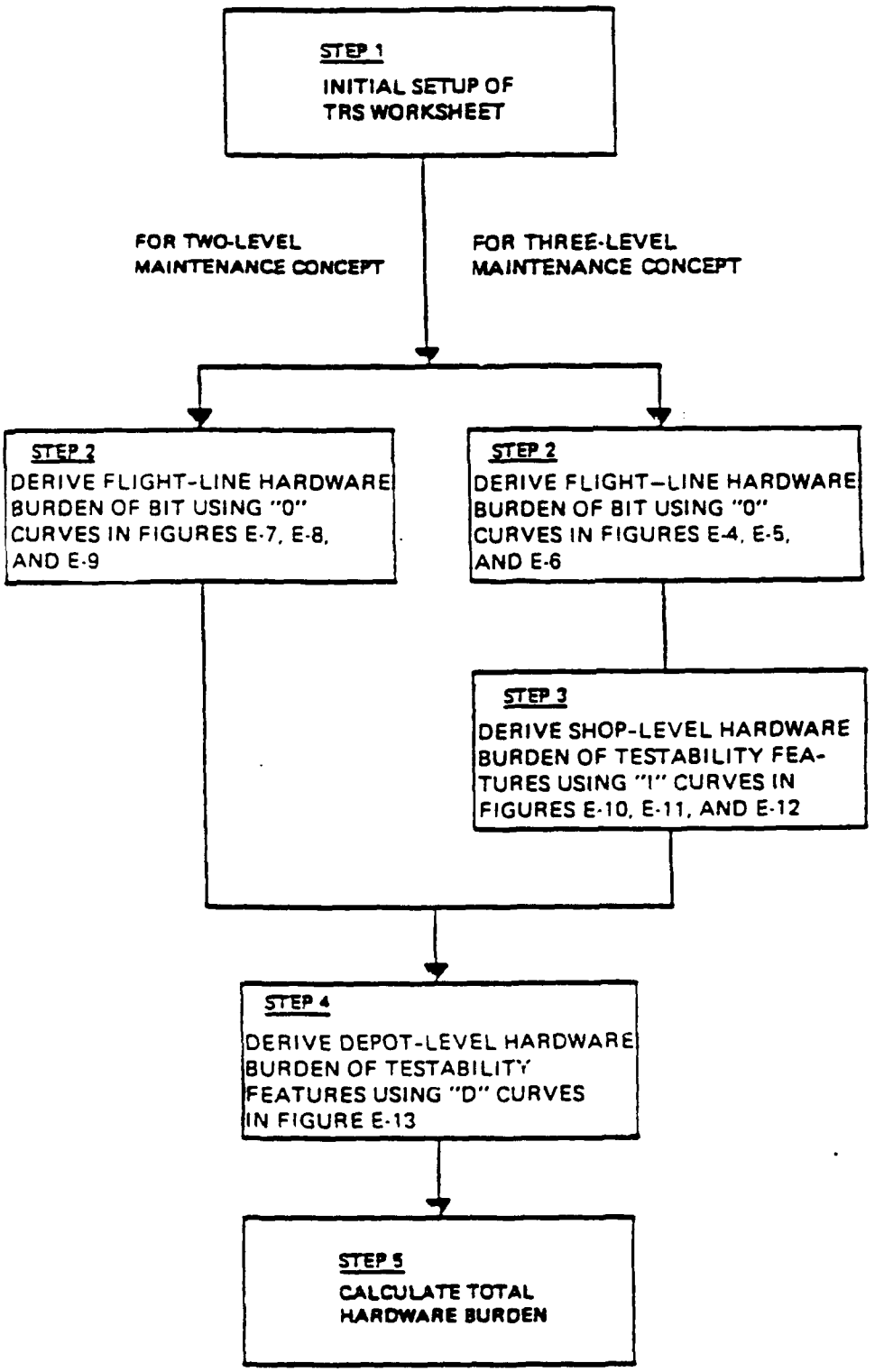


Figure E-1 Flow Diagram of Hardware Burden of BIT/BITE Testability Features Procedure (ATE GUIDE G3\3P2)

For the Determination of Avionics Subsystems
Compensated Burden Factor

Generic Category/Subsystem Name _____

Analog/Digital Mix Factor _____
Test Difficulty Factor _____
LRU Modularity Factors - Isolation To: 1 LRU _____
2 LRUs _____
3 LRUs _____

Flight Line - Hardware Burden of BIT/BITE Probability of Isolation _____	Three Level (LOR) Maintenance Program Fault Isolation To an AGS of:			Two Level (LOR) Maintenance Program Fault Isolation To an AGS of:		
Step	1 LRU	2 LRUs	3 LRUs	1 SRU	2 SRUs	3 SRUs
1 Uncompensated Burden Using Aid of BIT/BITE From "0" Curves Using A/D Mix Factor*	___	___	___	___	___	___
2 Multiply by Test Difficulty Factor	x ___	x ___	x ___	x ___	x ___	x ___
3 Adjusted Hardware Burden of BIT/BITE	___	___	___	___	___	___
4 Multiply by LRU Modularity Factors	x ___	x ___	x ___	x ___	x ___	x ___
5 Flight Line Hardware Burden of BIT/BITE	(A) ___	(B) ___	(C) ___	(D) ___	(E) ___	(F) ___

Shop Level - Hardware Burden of Testability Features Probability of Isolation _____	Fault Isolation To an AGS of:		
	1 SRU	2 SRUs	3 SRUs
6 Uncompensated Hardware Burden of BIT From "1" Curves Using A/D Mix Factor*	___	___	___
7 Multiply by Test Difficulty Factor	x ___	x ___	x ___
8 Shop Level Hardware Burden of BIT/BITE	(G) ___	(H) ___	(I) ___

Depot Level - Hardware Burden of Testability Features Probability of Isolation _____	Fault Isolation To an AGS of: 1 - (7) Components	Fault Isolation To an AGS of: 1 - (7) Components
9 Hardware Burden of Testability Features	(J) ___	(J) ___

10 Total Compensated Hardware Burden Factor %		
Isolation to 1 LRU/SRU	___ = (A) ___ + (G) ___ + (J) ___	___ = (D) ___ + (J) ___
Isolation to 2 LRUs/SRUs	___ = (B) ___ + (H) ___ + (J) ___	___ = (E) ___ + (J) ___
Isolation to 3 LRUs/SRUs	___ = (C) ___ + (I) ___ + (J) ___	___ = (F) ___ + (J) ___

Note: * – Selected data from Tables E-3 thru E-6 or from Figures E-7 thru E-16.

Figure E-2 Test Burden Worksheet (NATE GUIDE G3V3P2).

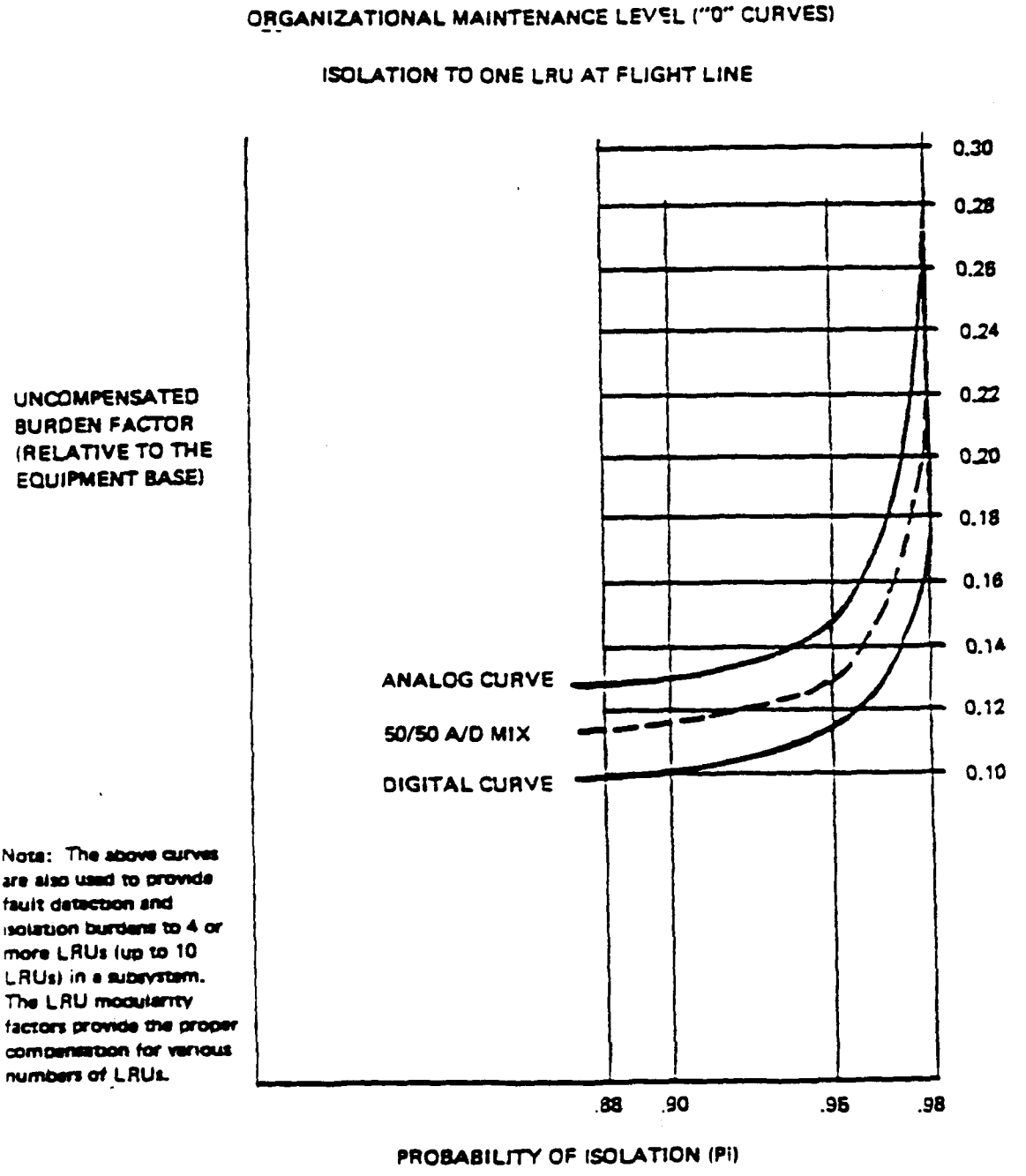


Figure E-4 Hardware Burden of BIT/BITE for Testability to Fault Isolate to One LRU at Flight Line (GATE GUIDE G3V3P2)

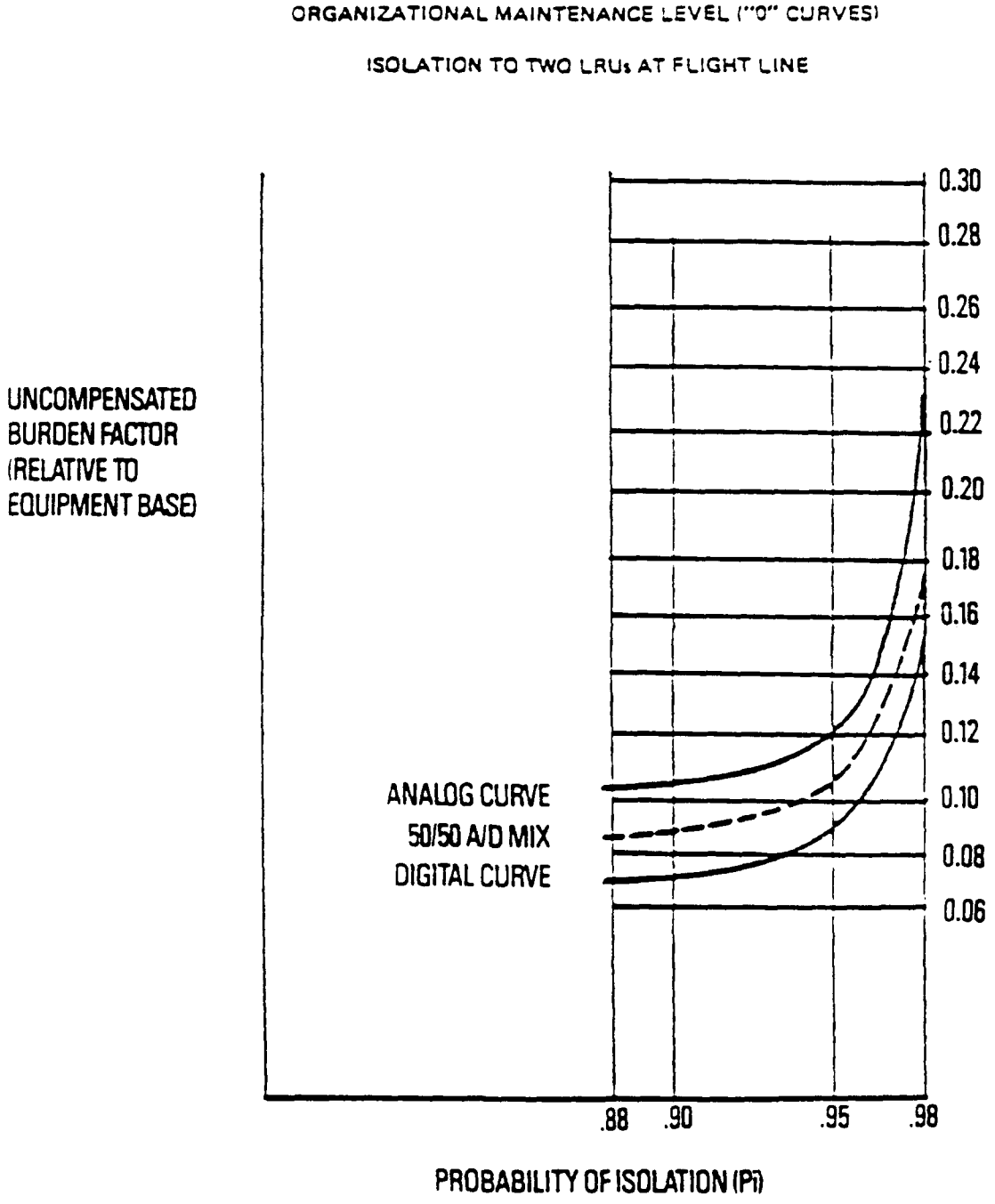


Figure E-5 Hardware Burden of BIT/BITE for Testability to Fault Isolate to Two LRUs at Flight Line (MATE GUIDE G3V3P2)

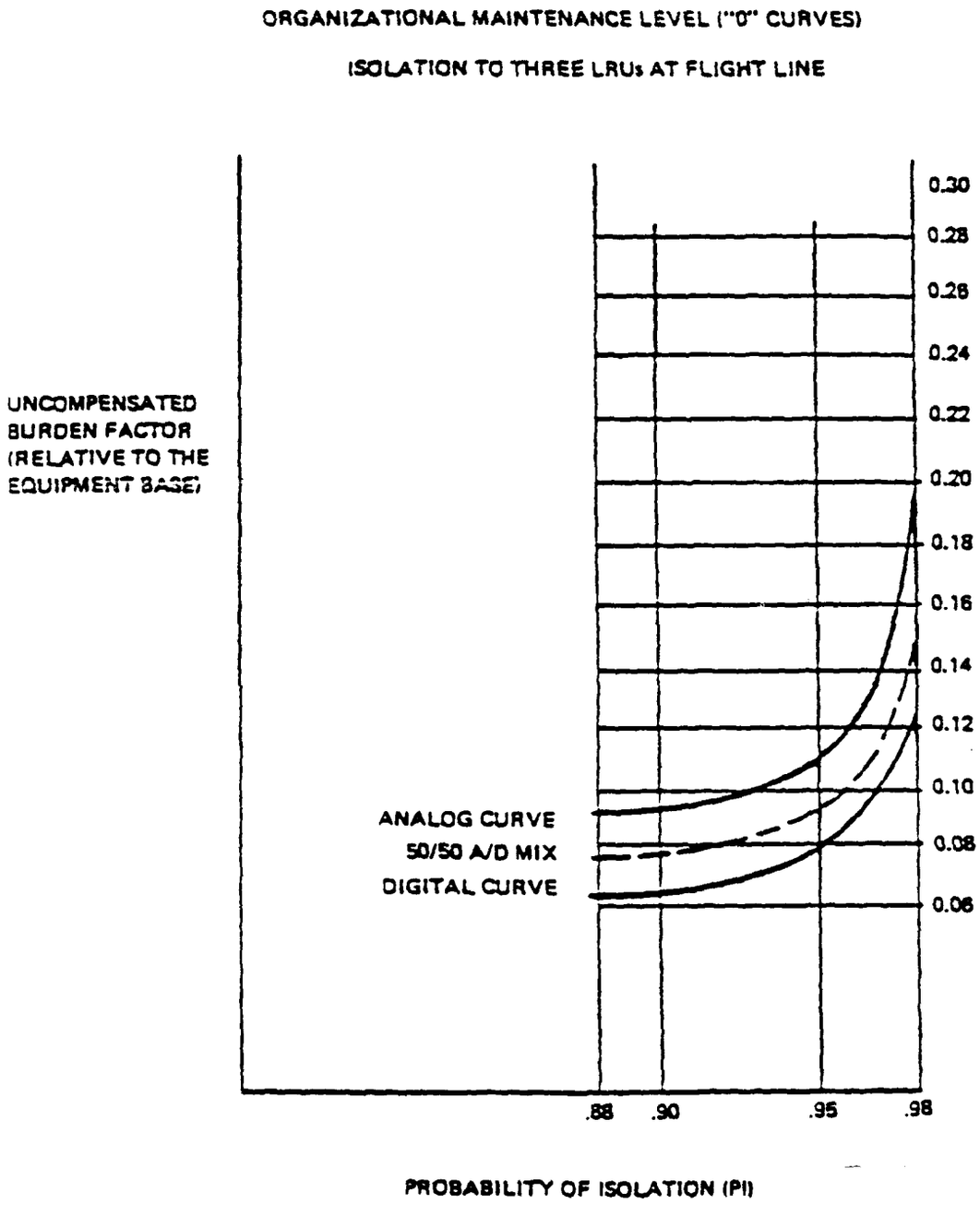


Figure E-6 Hardware Burden of BIT/BITE for Testability to Fault Isolate to Three LRUs at Flight Line (MATE GUIDE G3V3P2)

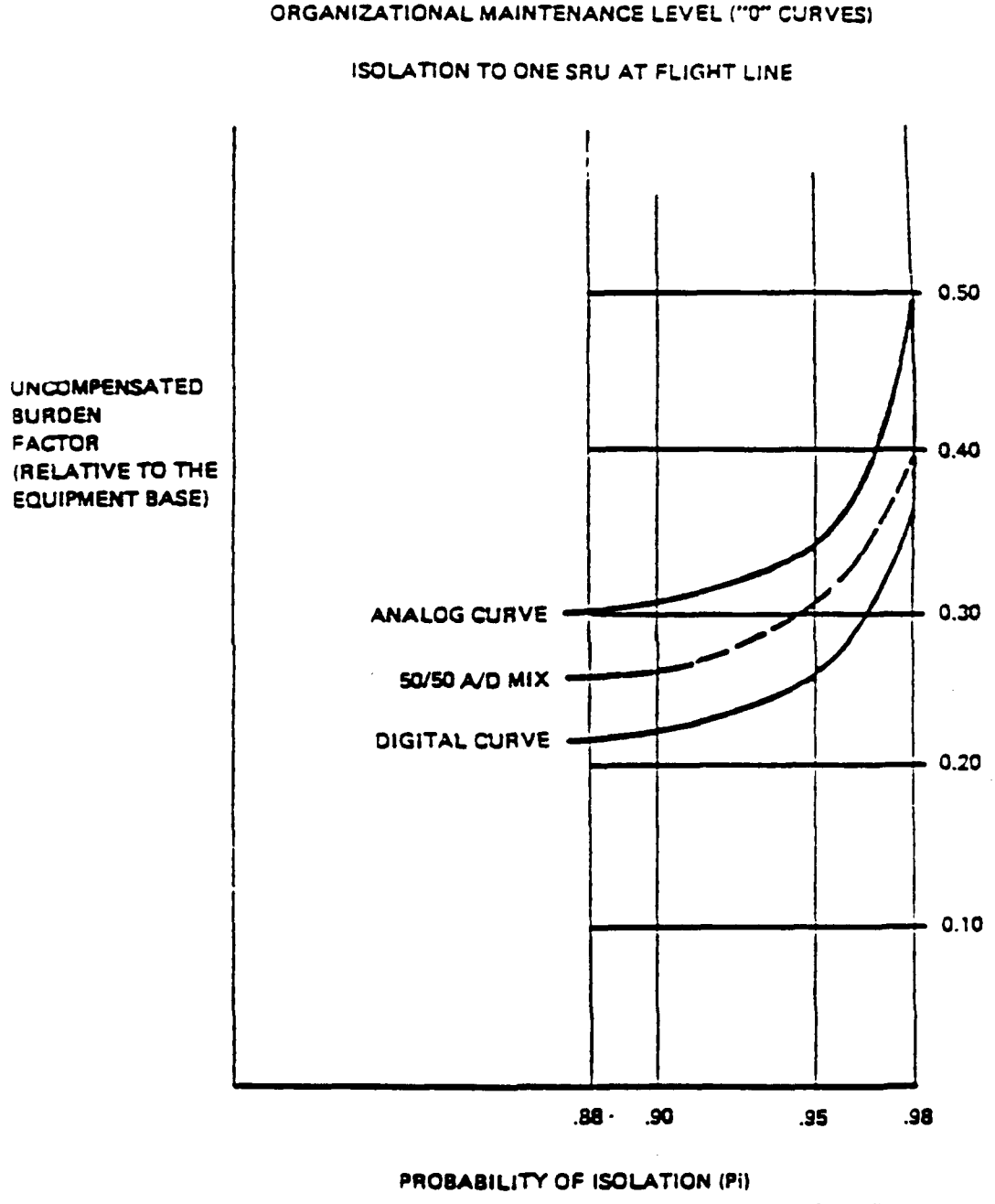


Figure E-7 Hardware Burden of BIT/BITE for Testability to Fault Isolate to One SRU at Flight Line (MATE GUIDE G3V3P2)

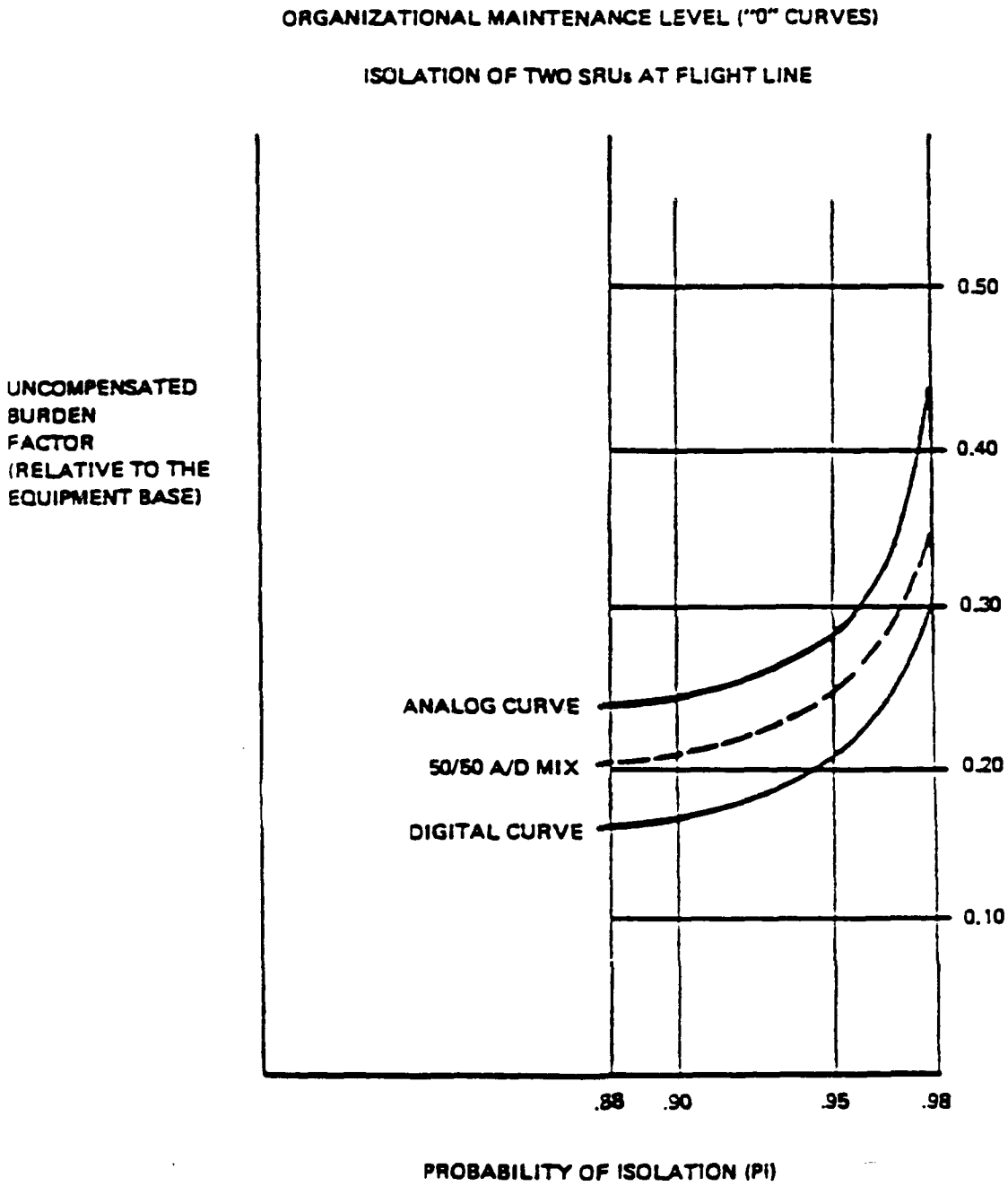


Figure E-8 Hardware Burden of BIT/BITE for Testability to Fault Isolate to Two SRUs at Flight Line (MATE GUIDE G3V3P2)

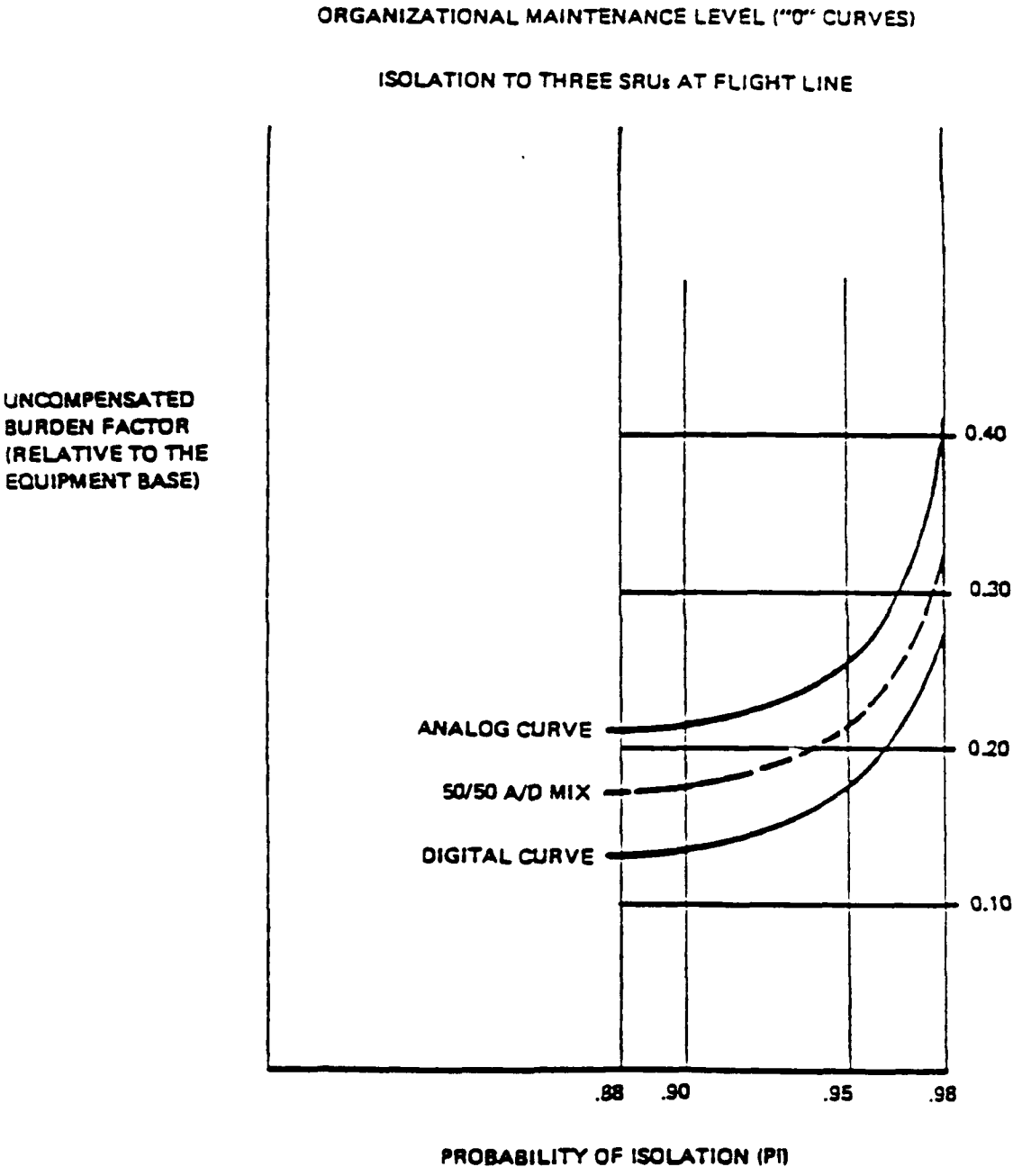
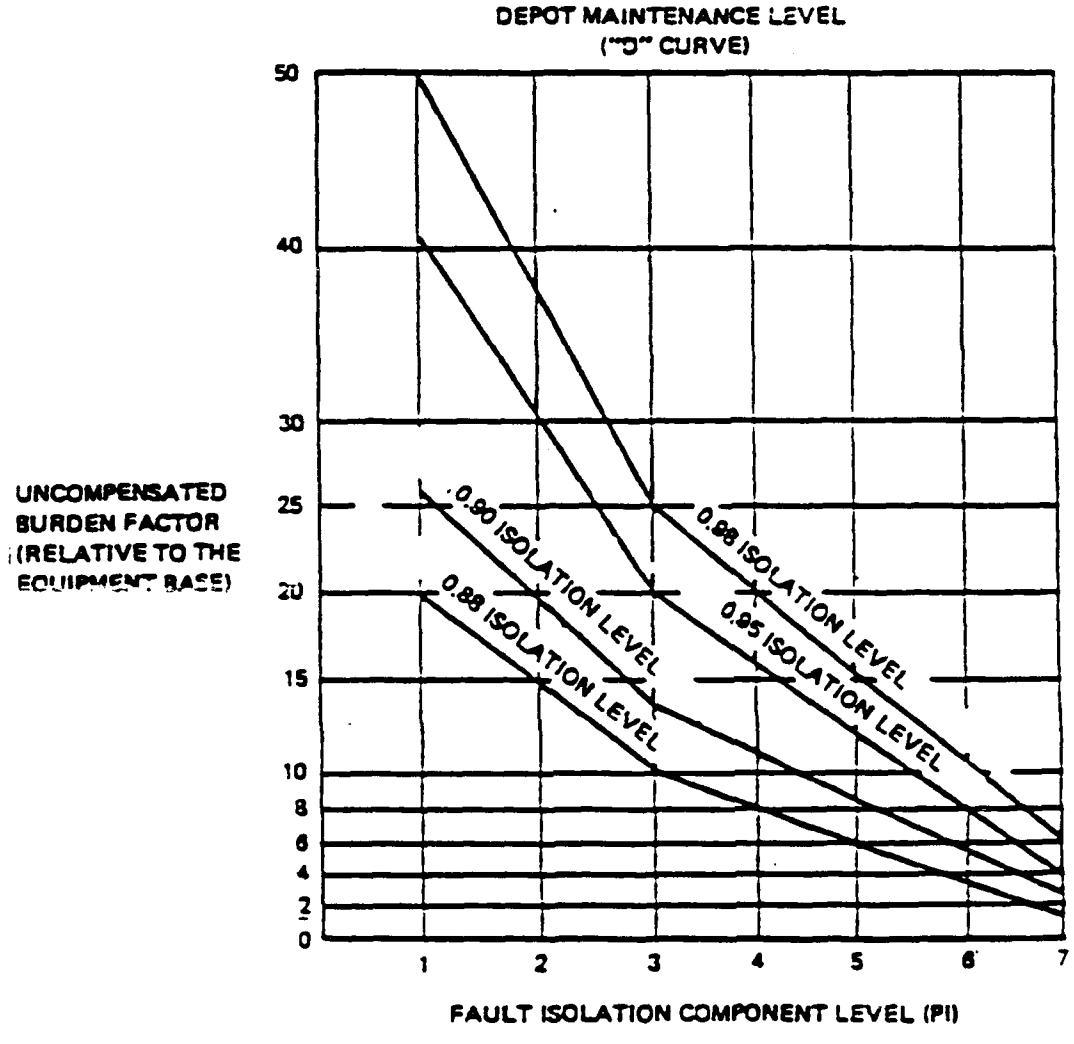


Figure E-9 Hardware Burden of BIT/BITE for Testability to Fault Isolate to Three SRUs at Flight Line (MATE GUIDE G3V3P2)



NOTE: HARDWARE BURDEN OF TESTABILITY FEATURES REPRESENTS THE BURDEN TO ACCESS AND BUFFER THE APPROPRIATE TEST POINTS.

Figure E-10 Hardware Burden of Testability Features for Testability to Fault Isolate to One to Seven Components on an SRU (MATE GUIDE G3V3P2 APPENDIX E, Figure E-13).

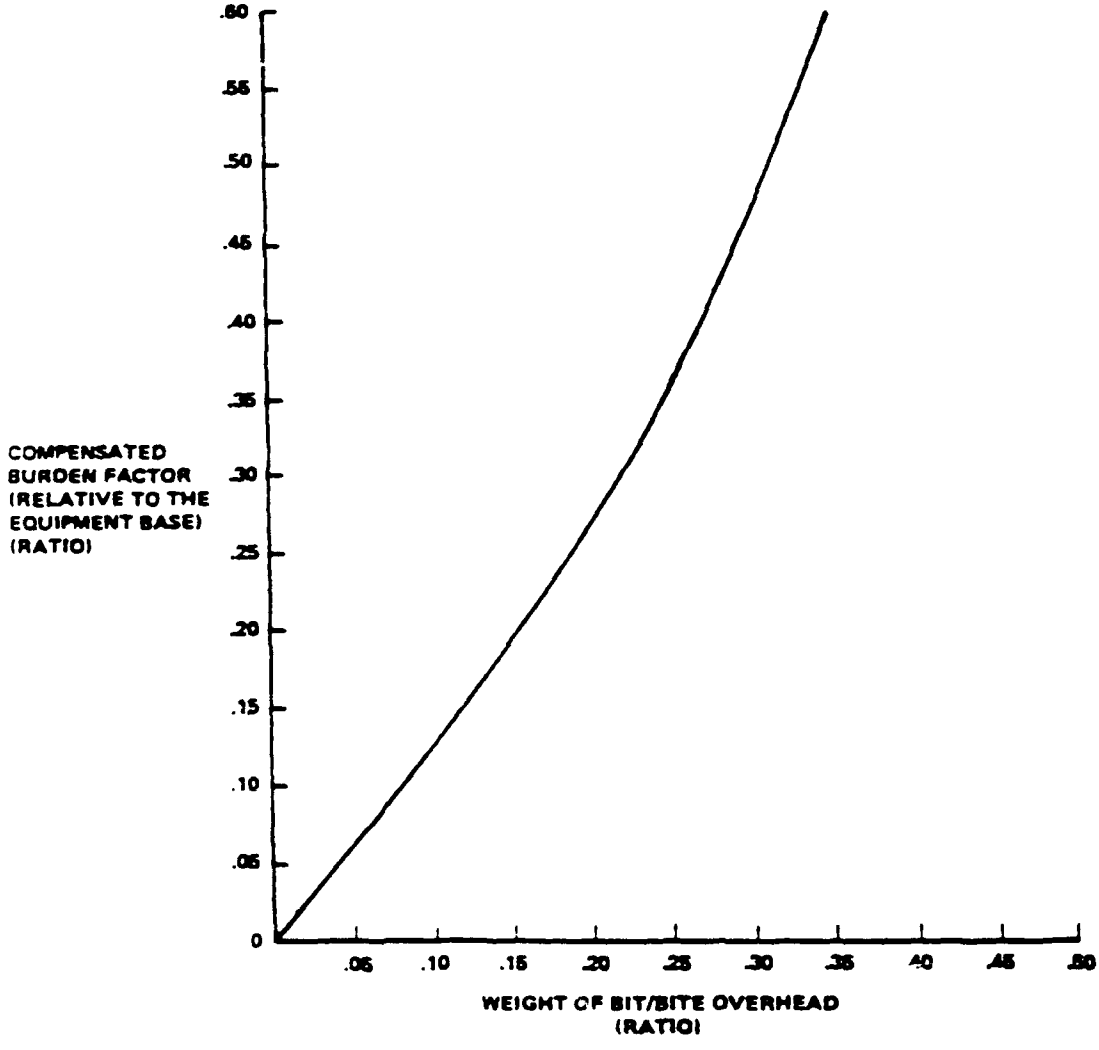


Figure E-11 Weight Overhead Factor vs. Compensated Burden Factor (MATE GUIDE G3V3P2S7, Figure 7-2).

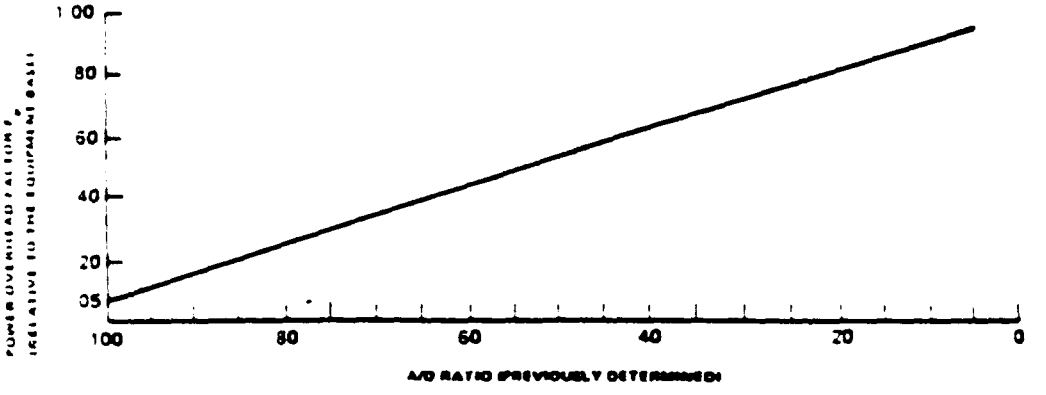


Figure E-12 Power Overhead Factor vs. A/D Ratio of BIT/BITE (MATE GUIDE G3V3P2S7, Figure 7-3).

GENERIC AVIONICS CATEGORY	AVIONIC EQUIPMENT GROUP	A/D MIX FACTOR	TEST DIFFICULTY FACTOR	Historic Number of LRUs
Communications	UHF	70/30	.9	2
	HF	70/30	.9	3
	VHF	70/30	.9	2
	Intercom	90/10	.4	4
	IFF	40/60	1	7
Navigation/Guidance	TACAN	40/60	1	2
	ADF/DME	80/20	.5	2
	AHRS	40/60	1.3	3
	Doppler Radar	30/70	1	3
	Inertial Nav	20/80	1.7	3
	Terrain Fol. Rdr.	40/60	2	5
	Radar Alt.	30/70	.9	2
	Search Radar	20/80	2	9
	ILS	40/60	.7	2
	Station Keeping	20/80	1	3
Fire Control/ Weapon Delivery/ Stores Management System	Attack Radar	30/70	2.0	8
	Lead Comp. Gyro	50/50	.9	2
	Gun Camera Elec	10/90	.5	2
	Arm. Cont. Set	20/80	1	3
	Digital Computer	20/80	1.5	2
	HUD	30/70	1.3	2
Penetration Aids	TEWS	50/50	1.3	4
Reconnaissance/Airborne Warning & Controls				
Flight Controls	AFCS	50/50	1.3	4
Auxiliary Electronics				
Central Integrated Checkout Avionics	CITS Computer	20/80	1.5	3
Instruments & Displays	Horiz. Sit. Disp.	50/50	4	2
	Vert. Sit. Disp.	50/50	.4	2
	Multipurpose Disp.	80/20	1.2	3
	Air Data Comp.	30/70	1.1	2
Bus	MIL-STD-1553	–	–	–
	Manchester Serial	–	–	–
	DAIS	–	–	–

TABLE T-1. Specific Testability Requirements (Generic)
(MATE GUIDE G3V3P2, Table E-1)

Number of LRUs in Subsystem	Number of LRUs to which fault isolation is desired									
	1	2	3	4	5	6	7	8	9	10
1	.20									
2	.33	.20								
3	.67	.33	.20							
4	.75	.50	.25	.20						
5	.80	.60	.40	.20	.20					
6	.83	.66	.50	.33	.20	.20				
7	.86	.73	.57	.43	.28	.25	.20			
8	.87	.75	.50	.50	.38	.25	.20	.20		
9	.89	.78	.66	.55	.44	.33	.22	.20	.20	
10	.90	.80	.70	.60	.50	.40	.30	.20	.20	.20

TABLE T-2 LRU Modularity Factors for Fault Isolation to 1-10 LRUs in a Subsystem. (MATE GUIDE G3V3P2. Table E-2).

CURVE	ORGANIZATIONAL						INTERMEDIATE			DEPOT
REFER TO FIGURE	E-4	E-5	E-6	E-7	E-8	E-9	E-10	E-11	E-12	E-13
AMBIGUITY GROUP SIZE / A/D FACTOR	1 LRU	2 LRUs	3 LRUs	1 SRU	2 SRUs	3 SRUs	1 SRU	2 SRUs	3 SRUs	7 COMPONENTS
100/000	12.9	10.4	9.1	30.0	23.8	21.4	3.5	3.2	3.0	1.4
90/10	12.7	10.1	8.8	29.2	23.2	20.6	3.3	3.0	2.9	1.4
80/20	12.4	9.8	8.6	28.4	22.4	19.8	3.2	2.9	2.8	1.4
70/30	12.1	9.5	8.3	27.6	21.6	19.0	3.1	2.8	2.7	1.4
60/40	11.8	9.2	8.0	26.8	20.8	18.2	3.0	2.7	2.6	1.4
50/50	11.5	8.9	7.7	26.0	20.0	17.4	2.9	2.6	2.5	1.4
40/60	11.2	8.6	7.5	25.2	19.2	16.6	2.8	2.5	2.4	1.4
30/70	10.9	8.3	7.2	24.4	18.4	15.8	2.7	2.4	2.3	1.4
20/80	10.6	8.0	6.9	23.6	17.6	15.0	2.6	2.3	2.2	1.4
10/90	10.3	7.7	6.6	22.8	16.8	14.2	2.5	2.2	2.1	1.4
000/100	9.9	7.4	6.4	21.8	16.0	13.4	2.4	2.1	2.0	1.4

TABLE T-3 Uncompensated Hardware Burden in Percent of Testability Features to Fault Isolate to Probability Level of .88 (MATE GUIDE G3V3P2, Table E-3).

CURVE	ORGANIZATIONAL						INTERMEDIATE			DEPOT
REFER TO FIGURE	E-4	E-5	E-6	E-7	E-8	E-9	E-10	E-11	E-12	E-13
AMBIGUITY GROUP SIZE / A/D FACTOR	1 LRU	2 LRUs	3 LRUs	1 SRU	2 SRUs	3 SRUs	1 SRU	2 SRUs	3 SRUs	7 COMPONENTS
100/000	13.0	10.6	9.2	30.5	24.0	21.7	3.6	3.3	3.1	2.8
90/10	12.8	10.3	9.0	29.6	23.3	20.9	3.4	3.1	3.0	2.8
80/20	12.5	10.0	8.8	28.7	22.6	20.1	3.3	3.0	2.9	2.8
70/30	12.2	9.7	8.5	27.8	21.9	19.3	3.2	2.9	2.8	2.8
60/40	11.9	9.4	8.2	26.9	21.2	18.5	3.1	2.8	2.7	2.8
50/50	11.6	9.1	7.9	26.0	20.5	17.7	3.0	2.7	2.6	2.8
40/60	11.3	8.8	7.6	25.2	19.6	16.9	2.9	2.6	2.5	2.8
30/70	11.0	8.5	7.3	24.5	18.7	16.1	2.8	2.5	2.4	2.8
20/80	10.7	8.2	7.0	23.7	17.9	15.3	2.7	2.4	2.3	2.8
10/90	10.4	7.9	6.8	23.0	17.1	14.5	2.6	2.3	2.2	2.8
000/100	10.0	7.6	6.6	22.2	16.3	13.7	2.5	2.2	2.1	2.8

TABLE T-4 Uncompensated Hardware Burden in Percent of Testability Features to Fault Isolate to a Probability Level of .90 (MATE GUIDE G3V3P2, Table E-4)

CURVE	ORGANIZATIONAL						INTERMEDIATE			DEPOT
REFER TO FIGURE	E-4	E-5	E-6	E-7	E-8	E-9	E-10	E-11	E-12	E-13
AMBIGUITY GROUP SIZE / A/D FACTOR	1 LRU	2 LRUs	3 LRUs	1 SRU	2 SRUs	3 SRUs	1 SRU	2 SRUs	3 SRUs	7 COMPONENTS
100/000	14.7	12.2	10.9	34.3	28.2	25.5	4.0	3.7	3.3	4.0
90/10	14.3	11.8	10.6	33.5	27.4	24.7	3.8	3.5	3.2	4.0
80/20	13.9	11.4	10.3	32.7	26.7	23.9	3.6	3.3	3.1	4.0
70/30	13.5	11.1	10.0	31.9	25.9	23.1	3.5	3.2	3.0	4.0
60/40	13.1	10.8	9.7	31.1	25.2	22.3	3.4	3.1	2.9	4.0
50/50	12.8	10.5	9.3	30.3	24.5	21.5	3.3	3.0	2.8	4.0
40/60	12.5	10.3	9.0	29.4	23.7	20.8	3.2	2.9	2.7	4.0
30/70	12.2	10.0	8.7	28.5	22.9	20.1	3.1	2.8	2.6	4.0
20/80	11.9	9.7	8.4	27.6	22.1	19.4	3.0	2.7	2.5	4.0
10/90	11.6	9.5	8.2	26.7	21.3	18.7	2.9	2.5	2.4	4.0
000/100	11.3	9.2	8.0	25.7	20.5	17.8	2.7	2.4	2.2	4.0

TABLE T-5 Uncompensated Hardware Burden in Percent of Testability Features to Fault Isolate to a Probability Level of .95 (MATE GUIDE G3V3P2, Table E-5)

CURVE	ORGANIZATIONAL						INTERMEDIATE			DEPOT
REFER TO FIGURE	E-4	E-5	E-6	E-7	E-8	E-9	E-10	E-11	E-12	E-13
A/D FACTOR \ AMBIGUITY GROUP SIZE	1 LRU	2 LRUs	3 LRUs	1 SRU	2 SRUs	3 SRUs	1 SRU	2 SRUs	3 SRUs	7 COMPONENTS
100/000	30.0	23.0	21.0	50.6	43.0	41.0	5.0	4.6	4.0	6.0
90/10	28.8	22.2	20.2	48.8	42.8	38.8	4.8	4.3	3.8	6.0
80/20	27.6	21.4	19.4	47.6	41.6	37.6	4.6	4.1	3.7	6.0
70/30	26.4	20.6	18.6	46.4	40.4	36.4	4.4	4.0	3.6	6.0
60/40	25.2	19.8	17.8	45.2	39.2	35.2	4.2	3.8	3.4	6.0
50/50	24.0	19.0	17.0	44.0	38.0	34.0	4.0	3.6	3.2	6.0
40/60	22.8	18.2	16.2	23.8	36.8	32.8	3.8	3.4	3.1	6.0
30/70	21.6	17.4	15.4	42.6	35.6	31.6	3.6	3.2	2.9	6.0
20/80	20.4	16.6	14.6	40.4	34.4	30.4	3.4	3.1	2.8	6.0
10/90	19.2	15.8	13.8	39.2	33.2	29.2	3.2	2.9	2.7	6.0
000/100	17.8	15.0	13.0	38.0	32.0	28.0	3.0	2.8	2.6	6.0

Table T-6 Uncompensated Hardware Burden in Percent of Testability Features to Fault Isolate to a Probability Level of .98 (MATE GUIDE G3V3P2, Table E-6)